FLOW CONTROL OF REAL TIME MULTIMEDIA APPLICATIONS USING MODEL PREDICTIVE CONTROL WITH A FEED FORWARD TERM

A Thesis

by

THIEN CHI DUONG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2010

Major Subject: Mechanical Engineering

Flow Control of Real Time Multimedia Applications Using Model Predictive Control

with Feed Forward Term

FLOW CONTROL OF REAL TIME MULTIMEDIA APPLICATIONS

USING MODEL PREDICTIVE CONTROL

WITH A FEED FORWARD TERM

A Thesis

by

THIEN CHI DUONG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Alexander G. Parlos
Committee Members,	Won-Jong Kim
	Jose Silva-Martinez
Head of Department,	Dennis O' Neal

December 2010

Major Subject: Mechanical Engineering

ABSTRACT

Flow Control of Real Time Multimedia Applications Using Model Predictive Control with a Feed Forward Term. (December 2010)

Thien Chi Duong, B.S, University of Technology at Ho Chi Minh City

Chair of Advisory Committee: Dr. Alexander G. Parlos

Multimedia applications over the Internet are getting more and more popular. While non-real-time streaming services, such as YouTube and Megavideo, are attracting millions of visiting per day, real-time conferencing applications, of which some instances are Skype and Yahoo Voice Chat, provide an interesting experience of communication. Together, they make the fancy Internet world become more and more amusing. Undoubtedly, multimedia flows will eventually dominate the computer network in the future.

As the population of multimedia flows increases gradually on the Internet, quality of their service (QoS) is more of a concern. At the moment, the Internet does not have any guarantee on the quality of multimedia services. To completely surpass this limitation, modifications to the network structure is a must. However, it will take years and billions of dollars in investment to achieve this goal. Meanwhile, it is essential to find alternative ways to improve the quality of multimedia services over the Internet.

In the past few years, many endeavors have been carried on to solve the problem. One interesting approach focuses on the development of end-to-end congestion control

strategies for UDP multimedia flows. Traditionally, packet losses and delays have been commonly used to develop many known control schemes. Each of them only characterizes some different aspects of network congestion; hence, they are not ideal as feedback signals alone. In this research, the flow accumulation is the signal used in feedback for flow control. It has the advantage of reflecting both packet losses and delays; therefore, it is a better choice. Using network simulations, the accumulations of real-time audio applications are collected to construct adaptive flow controllers. The reason for choosing these applications is that they introduce more control challenges than non-real-time services.

One promising flow control strategy was proposed by Bhattacharya and it was based on Model Predictive Control (MPC). The controller was constructed from an ARX predictor. It was demonstrated that this control scheme delivers a good QoS while reducing bandwidth use in the controlled flows by 31% to 44%. However, the controller sometime shows erratic response and bandwidth usage jumps frequently between lowest and highest values. This is not desirable. For an ideal controller, the controlled bandwidth should vary near its mean. To eliminate the deficiency in the strategy proposed by Bhattacharya, it is proposed to introduce a feed forward term into the MPC formulation, in addition to the feedback terms. Simulations show that the modified MPC strategy maintains the benefits of the Bhattacharya strategy. Furthermore, it increases the probability of bandwidth savings from 58% for the case of Bhattacharya model to about 99% for this work.

To my beloved parents and brother

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

A. Research Motivation

Online multimedia applications have been quickly attracting the interest of the networking community. These visual ways of sharing individual information is so convenient that, perhaps, they tend to dominate the Internet in near future. Quality of such services, hence, becomes more and more important. Unfortunately, the current best effort network offers no guarantee on such quality of services. In fact, it is originally designed for services using TCP protocol while multimedia applications prefer UDP protocol for its efficiency of data transferring. Certainly, structural modification of computer network is the ultimate solution for the emerging trend. However, this will be time and money consuming. Therefore, finding an alternative way to improve the quality of multimedia services has become the common objective of many researches. Among them, a promising approach is to embed into UDP some type of control, which is similar to the case of TCP.

B. Literature Review

1. Prior Literature

There have been numerous studies focused on modeling, prediction and control of

---

This thesis follows the style of *IEEE Transactions on Automatic Control.*

network traffic. For example, Shah et al. [1] utilize linear and nonlinear AR models along with state-space techniques to compare their prediction to a simple predictor, a predictor that assumes the current actual value to be future value. They suggest that prediction at larger time scales is more promising than at smaller time scales. Ohsaki et al. [2], [3] model the Internet using the Autoregressive Exogenous (ARX) technique. They choose packet inter-departure time from the source as the input and round-trip time variation as the output. They show that an ARX model is suitable for Local Area Networks (LAN) and also for Wide Area Networks (WAN), should the bottle neck link be shared by a small number of users.

Parlos [4] suggests using neuro-predictors to perform multi-step-ahead prediction of network delay. Even though the proposed multi-step predictions may be inaccurate compared to single-step ones, they are useful in term of improving the QoS. Wang et al. [5] compare Radial Basis Functions (RBF) to linear predictors in predicting network delays. They find that once the RBF network out performs the linear predictors when it is trained sufficiently. Jiang et al [6] develop a “model-free” fuzzy time-series predictor for predicting packet arrival patterns for multi-media traffic. They verify their model using data obtained from a simulated continuous-state autoregressive Markov model along with a ‘Star Wars’ video-traffic data set. Doddi [7] develop auto regressive and neural networks based predictors and analyze them for various ns-2 simulation scenarios. Edmund et al [8] use back propagation based feed forward neural network to perform time series prediction of network delays. They conclude that their neural network is an attractive alternative to traditional regression techniques. However, unreliable training and test data, obtained using simulation of an AR Markov model, is the shortcoming of the work.

Bhattacharya [9], [10] believes that a neural network does not out-perform an ARX predictor. He proposes a Model Predictive Control (MPC) scheme based on ARX predictor and utilizes sending bit rate as input and accumulation as output signal. The control effort is to remain the accumulations of real-time multimedia flows at a certain reference. From network simulation results, he concluded that the control model delivers a good QoS while reducing bandwidth use of controlled flows by 31.43% to 43.96%.

2. Background

a. Voice Encoder-Decoder and Network Input

The Internet uses packet switching as its delivery method. By this method, video or audio data is divided into small packets before being sent to its receiver. The size of these packets determines the application's sending bit rate (or bandwidth) and is not arbitrary. In fact, the division process is managed by a program called encoder which has a certain underlying algorithm. This algorithm can only allow data to be divided into several certain sizes. Some commercially available encoders include: G.729a, AMR, iLBC, Speex… Usually, these encoders can generate one or two different sending bit rates. As a counter part of encoder, a corresponding decoder will combine received packets to re-construct the original data at the receiver.

In this research, the encoder recommended by Bhattacharya [9] is used. This is a theoretical encoder that is assumed to generate six of the following bit rates:

1. 36 kbps: 90 byte packets (including all headers) sent every 20ms.
2. 48 kbps: 120 byte packets (including all headers) sent every 20ms.
3. 60 kbps: 150 byte packets (including all headers) sent every 20ms.
4. 72 kbps: 180 byte packets (including all headers) sent every 20ms.
5. 84 kbps: 210 byte packets (including all headers) sent every 20ms.
6. 96 kbps: 240 byte packets (including all headers) sent every 20ms.

These sending bit rates will serve as the input signal throughout the processes of system identification, predictor construction, and control design in this research.

b. Accumulation

The effort is to control the congestion of real-time audio conferencing flows. Therefore, the output signal must characterize flow congestion. Traditionally, packet losses and delays play a significant role. However, each of them can only characterize one part of network congestion; hence, they are not good as output signals alone. Xia et al [11] suggested using the accumulation of bytes in the network to measure network congestion. Formally defined, accumulation is the difference between the number of bytes sent into the network by the source and the number of bytes received by the destination at time *t*. This signal has the advantage of reflecting both packet losses and the effect of delays within the network. Hence, it is a better choice for characterizing network congestion.

c. Network Simulator ns-2

To test the efficiency of the proposed control strategies, experiments are carried on in a network simulation environment. There are two popular network simulators which are ns-2 and Opnet. Here, ns-2 is chosen for the task. Ns-2 is a discrete event network simulator written by researchers at UC Berkeley. It provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless networks. It is a commonly used simulator for networking researchers because it is open source and widely accepted in networking community. Details on ns-2 can be found at [12].

d. Comprehensive Loss Rate

In computer networking, packet loss terminology means packets that are dropped off in queues of the intermediate nodes while traversing through the network. This is considered as pure loss. However, it is not the only kind of loss in real time multimedia. Psychologically, users will experience unpleasant telephonic conversation if the packet arrives late after some deadline threshold. These packets are considered to be lost as well. Delay loss would be a suitable name to feature this type of packet loss. The comprehensive loss for real time multimedia application is the sum of pure and delay packet loss. Consequently, the comprehensive loss rate is defined as the ratio of the number of comprehensive loss packets and the total number of packets sent from server to client at time interval $[t_1, t_2)$.

e. Voice Quality Measurement

Comprehensive loss rate is one criterion to construct different experimental topologies for designing and testing purposes within the current research. It can also be used as a metric for judging the control strategy's efficiency. However, it is not a best measurement for evaluating QoS of real-time audio conferencing applications. Instead, the voice quality can be evaluated by Mean Opinion Score (MOS). MOS is a scale that provides a quantitative estimate of the perceptual voice quality. MOS tests for voice are specified by ITU Telecommunication Standardization Sector (ITU-T) [13]. The MOS scale varies from 1 to 5, with 1 indicating worst quality audio and 5 representing excellent quality audio without any impairments due to encoder and decoder. By conventional standards of Plain Old Telephone System (POTS), a "toll quality" telephone service has a MOS level of 4.0 or above.

MOS can be calculated as follow:

$$MOS = 1 + 0.035R + 7\times10^{-6} R\left(R-60\right)(100-R), \tag{1.1}$$

where: R is called R-factor that combines different aspect of voice quality impairments and is defined by E-model as:

$$R = 94.5 - I_e - I_d\,, \tag{1.2}$$

where: I_d depends on mouth to ear delay, d, which is the sum of the end-to-end delay of the packet containing the encoded voice data and the delay due to coding and decoding of the signal:

$$I_d = 0024d + 0.11\left(d - 177.3\right)I(d - 177.3), \tag{1.3}$$

with:

$$I(x)=\begin{cases}0, & if \;\; x<0\\ 1, & otherwise\end{cases}$$

I_e depends on parameters that are determined by the properties of the encoder. The relation between I_e and overall packet loss rate e is expressed in following equation:

$$I_e = \gamma_1 + \gamma_2 \ln(1+\gamma_3 e), \tag{1.4}$$

where: γ_1 is the constant that determines voice quality impairmentcased by encoding. γ_2 and γ_3 describe the impact of loss on perceived voice quality for a given codec. e includes both network losses and playout buffer losses.

More specific definitions and calculations of related parameters can be found in Bhattacharya [9].

C. Problem Definition

Inspired by TCP, the current research aims to improve the QoS of multimedia applications by inducing novel adaptive end-to-end flow control strategies into UDP applications. Here, the interest falls upon audio conferencing applications. Since real-time multimedia applications involve more difficulties than non-real-time multimedia services, strategies designed for the latter could not be expected to work for the former. Network simulator ns-2, written at UC Berkeley, is used to simulate best effort networks. Experiments are carried on various simulation scenarios from narrowing the backbone bandwidth to increasing the number of UDP flows.

D. Research Objectives

The objectives of this research are:

- Develop predictive control schemes for real-time multimedia applications.
- Demonstrate the contribution of the proposed predictive control strategies in keeping the QoS of real-time multimedia applications high while introducing a friendly network environment that reduces bandwidth usage.

E. Proposal Approach

The approach proposed for utilization in the current research is:

- The real-time multimedia flow controller proposed by Bhattacharya is used. However, the controller sometime shows erratic responses of which bandwidth usage jumps frequently between lowest and highest values. This makes the control strategy less desirable; an ideal controller should have its bandwidth usage varied near the bandwidth's mean.
- In this work, the deficiency of the MPC controller proposed by Bhattacharya is eliminated by the use of a feed forward term in addition to the feedback terms.

F. Contribution of the Thesis

The anticipated contributions of this research are:

- To improve the performance of the MPC controller proposed by Bhattacharya by eliminating the large variation in the send rates observed.
- To develop a flow controller that maintains the QoS of real-time multimedia applications while lowering their use of network bandwidth.

G. Organization of the Thesis

The thesis has been divided into six chapters. The first chapter gave an introduction about online multimedia application, its potential proliferation over the Internet and the proposal for improving its quality of services. Literature review and background are included. Chapter II describes the experimental setups and simulation scenarios used in the thesis. A brief review of linear system identification is given in Chapter III. In particular, the ARX model is at interest. Chapter IV dedicates to the model predictive control theory and its formulations based on ARX models. In Chapter V, experiment results will be shown and analyzed to see the efficiency of the designed control schemes. Finally, Chapter VI will conclude the work.

CHAPTER II

SIMULATION SETUP

Best experimental environment for the research should be the real world platform. However, it is not easy to carry experiments on the Internet. PlanetLab, which is a globally distributed open platform for developing, deploying and accessing planetary-scale network services, seems to be a logical solution. Nevertheless, the experimental data collected from PlanetLab overlay networks failed to fulfill the expectations of a real world test bed. This is because of the inability of PlanetLab in providing the right level of network congestion. More about PlanetLab failure can be found in [1].

The limitations of PlanetLab direct the need for a controllable experimental platform to network simulators. Even though network simulators are not real (physical) test beds, they allow user to have a full control over all aspects of simulated networks. This sufficiently saturates the requirements of the current research. Here, ns-2, written at UC Berkeley, is chosen to simulate two different network topologies.

As mentioned, this research aims to improve the MPC strategy proposed by Bhattacharya by adding a feed forward term in addition to feedback terms. To see how effective the proposed control strategy, compared to Bhattacharya's strategy, is, two different network topologies used in Bhattacharya's work are at concern: (1) one for designing and validating flow control strategies and (2) one for studying the scalability of the designed control strategies.

A. Simulation Topology for Designing and Validating Flow Control Strategies

The first topology for designing and testing of control algorithms is show in Figure 2.1. As can be seen, the topology has two backbone links. The first link is a duplex connection of routers R0 and R1 while the second duplex link is formed by routers R1 and R2.

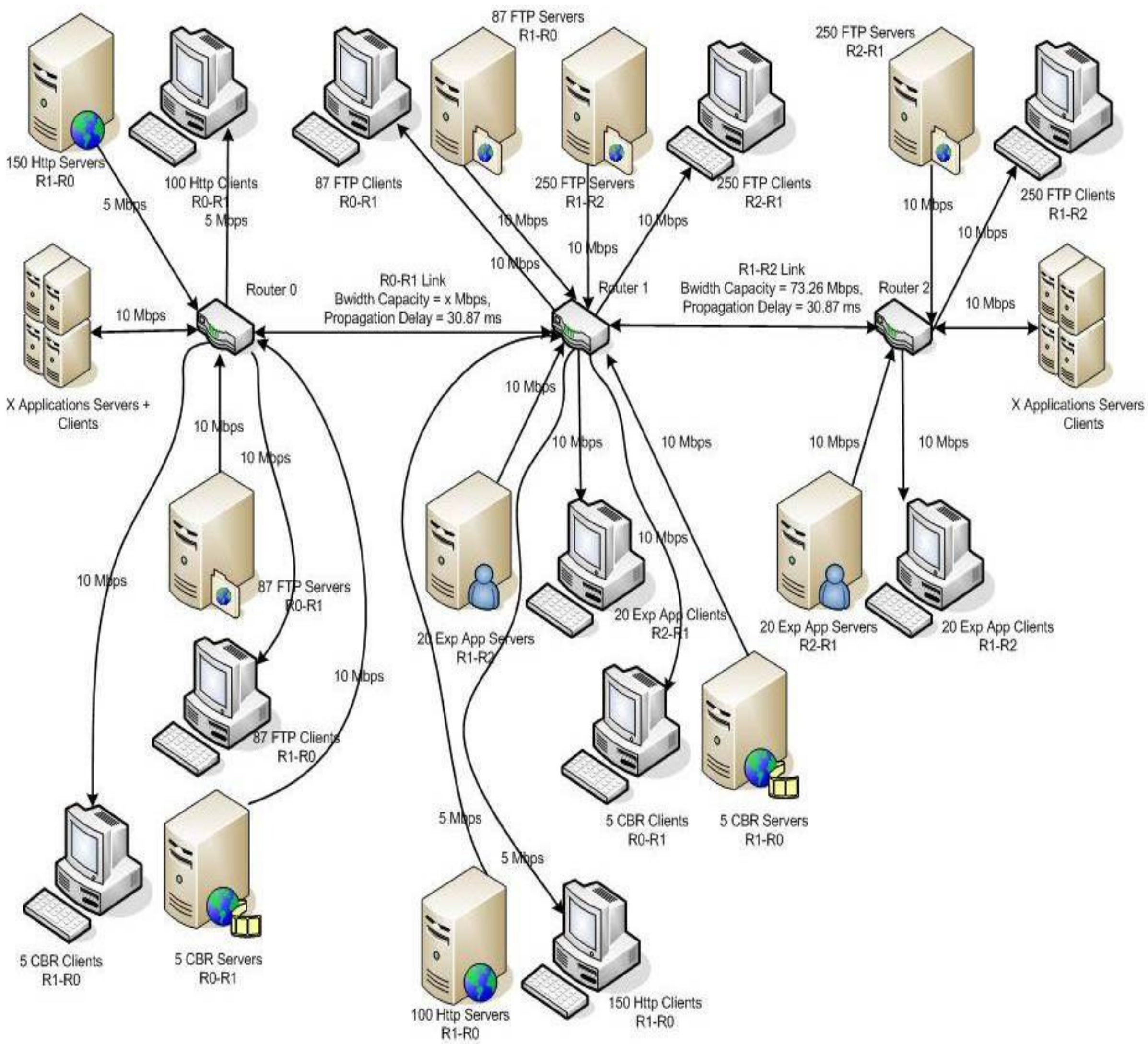

Fig. 2.1. Topology #1 for designing and validating of control strategies.

Important parameters of these backbones include propagation delays, bandwidths and queue sizes:

(1) Propagation delays: To resemble the propagation delays in United States which vary from 60 ms to 90 ms, the propagation delays of two backbones are chosen to be 30.87 ms. Hence, their total propagation delay adds up to 61.74 ms.

(2) Bandwidths: Bandwidth of the second link (R1-R2) is chosen to be fixed at 73.26 Mbps while the first link (R0-R1) varies from 28.93 Mbps to 50.92 Mbps. The purpose of this variation is to create six different loss rate scenarios for building and validating of the designed control strategies. The bandwidth capacities of link (R0-R1) and their corresponding average comprehensive loss rates (CLR) of five observed UDP flows are shown in Table I.

(3) Queue sizes: two queues in the first duplex link R0-R1 have the size of 460800 bytes. The next duplex link R1-R2 has queues with capacities of 1024000 bytes. All of these queues follow First In First Out (FIFO) buffer management scheme.

Table I. R0-R1's bandwidth capacities and corresponding CLRs.

	Bandwidth capacities of link R0-R1 (Mbps)	CLR (%)
1	50.92	3
2	44.92	5
3	40.92	7
4	36.92	9
5	35.92	11
6	28.92	15

Along the backbones, different network components are assigned. Considering left to right as the forward direction in Figure 2.1, the forward cross-flow traffic on link R0-R1 consists of 150 HTTP (TCP), 5 CBR (UDP) and 87 FTP (TCP) flows while there are 100 HTTP (TCP), 5 CBR (UDP) and 87 FTP (TCP) flows making up the backward cross-flow traffic. The HTTP nodes use 5 Mbps capacity links to connect to router R0 and router R1; other nodes are connected to R0 and R1 by 10 Mbps capacity links. The cross-flow traffic on link R1-R2 comprises of 250 FTP (TCP), and 20 Exponential flows (UDP) in both forward and backward directions. All of these flows are connected to router R1 and R2 using 10 Mbps links.

The above cross-traffic flows constitute 98.98 % of the total flows. The remaining 1.02% flows are real-time audio conferencing applications using UDP protocol. These UDP flows will be manipulated by the MPC strategies. Within this topology, five controlled UDP are connected to router R0 and another five corresponding controlled UDP are connected to router R2. The links between these nodes and routers are in 10 Mbps capacity. It is easy to notice that the controlled UDP flows traverse through the network backbones.

The contributions of each type of flows in term of percentage of traffic bytes are given in Table II. Even though the comprehensive loss rate of the controlled UDP varies from 3% to 15% due to the variation of link R0-R1's bandwidth, the relative contribution of each type of traffic merely changes within a bound of ±1%.

Table II. Details of the composition of cross traffic in terms of flows for the network topology used to design and validate the control schemes.

Type	% of Total TCP	% of Total UDP	% of Total Traffic
TCP (HTTP)	~ 15.69	N.A.	~ 12.64
TCP (FTP)	~ 84.31	N.A.	~ 67.92
Total % (TCP / Total Traffic)			~ 80.56
Controlled UDP	N.A.	~ 2.98	~ 0.58
UDP (Exp)	N.A.	~ 84.30	~ 16.39
UDP (CBR)	N.A.	~ 12.72	~ 2.47
Total % (UDP / Total Traffic)			~ 19.44

The simulations of the first network topology at different loss rates varying from 3% to 15% are for collecting data to model this topology. During each simulation, packets are sent from one end to the other every 20 ms. The packet size (or bit rate) switches randomly every 240 ms among six different levels, which are generated by the recommended encoder [9]. The corresponding instance accumulation signals are measured every 20 ms and averaged over the period of 240 ms. The mean bit rates and accumulations together form the input-output pairs needed for modeling the network topology.

B. Simulation Topology for Studying the Scalability of the MPC Strategies

Figure 2.2 shows the network topology for studying the scalability of the MPC strategies. The purpose of this experiment is to see what happens when the number of controlled real-time multimedia applications grows up in the network. Two different circumstances of this increasing are analyzed:

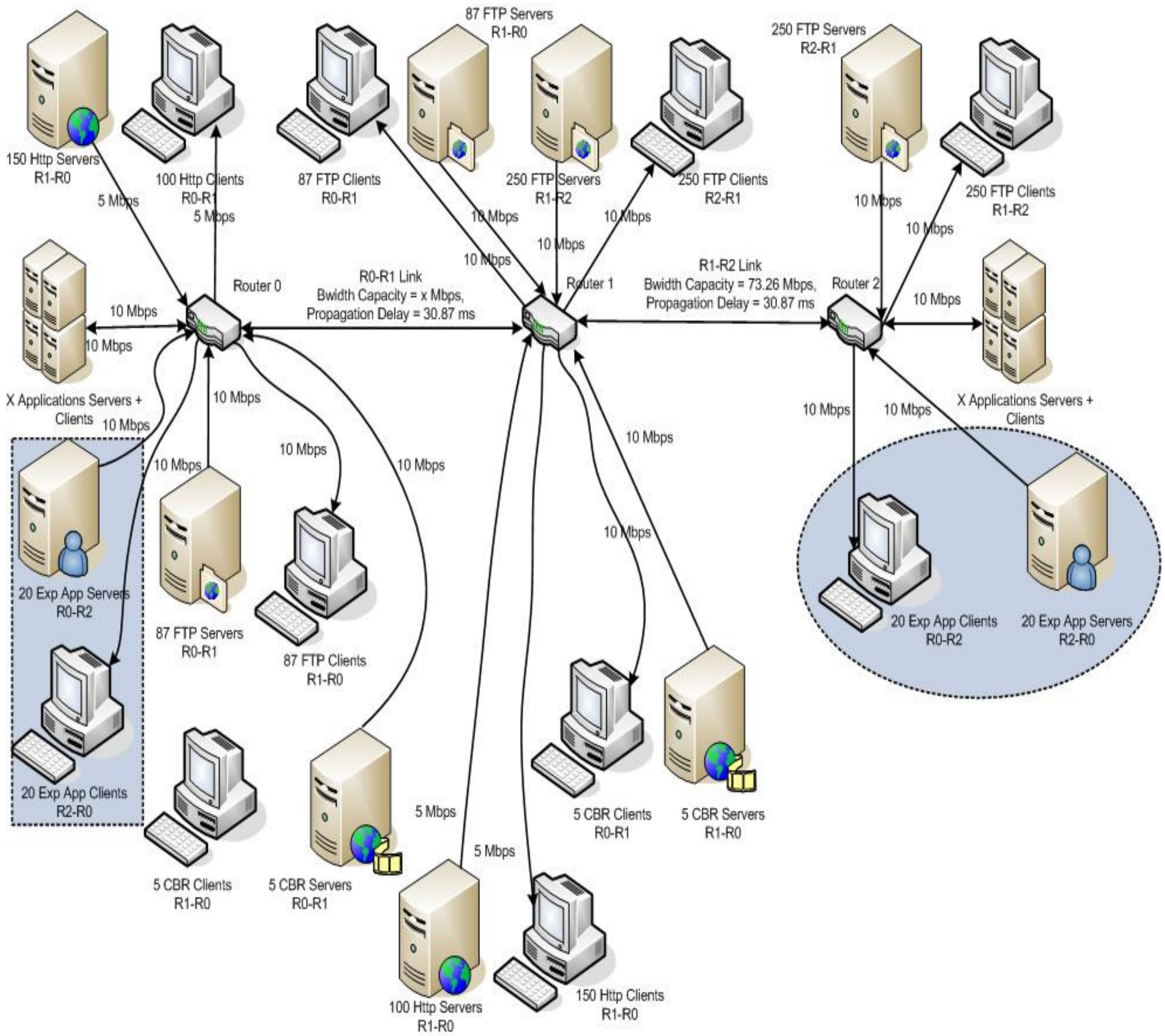

Fig. 2.2. Topology #2 for studying scalability of MPC strategies.

(1) The share of controlled UDP flows to the total UDP traffic increases while the ratio of UDP flows and TCP flows to the total traffic remains the same.

(2) The share of total UDP flows to the total traffic grows up.

These two circumstances are shown in Figure 2.3.

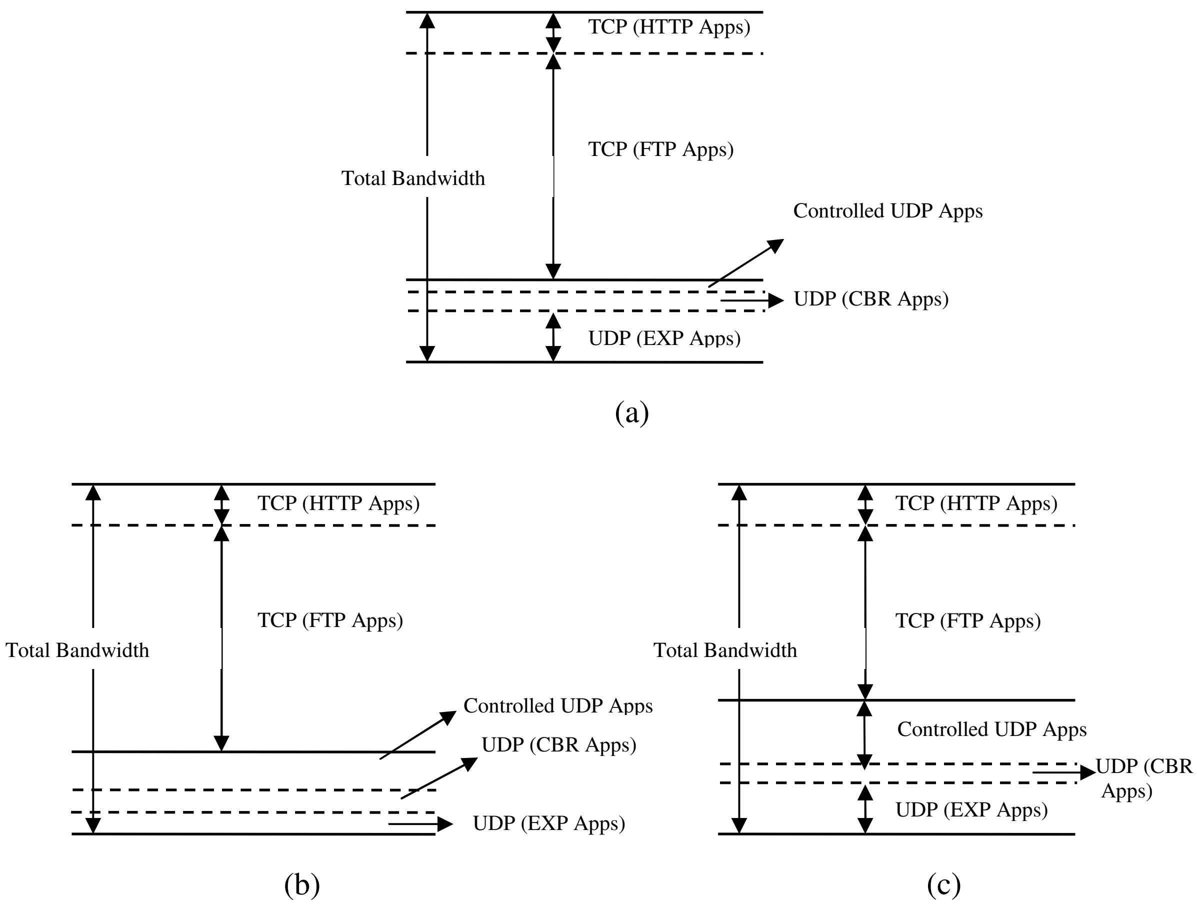

Fig. 2.3. Objectives of the scalability experiments.

The featured difference between current and previous topology is that 20 Exponential (UDP) flows are moved from router R1 to router R0. This rises up some robustness for the testing of the designed controllers. In addition, Bhattacharya intentionally placed all HTTP and UDP flows under same routes to see the impact of adding more controlled UDP flows on the QoS of HTTP services, the dominating services in the Internet.

The moving of Exponential nodes to router R0 requires the link R0-R1 to be able to tolerate extra traffic. Therefore, the queue sizes of the duplex link R0-R1 are increased to 768000 bytes from 460800 bytes. In addition, the link's bandwidth capacity is expanded to 75.92 Mbps and kept fixed since the effect of different loss rates, which has already been considered in the first topology, is not at particular interest in this topology.

To study the scalability of the MPC strategies under the first circumstance, four sets of experiments are carried on. In each set, the average bit rate of exponential (UDP) nodes is alternatively reduced. The gaining bandwidth is allocated to newly added UDP nodes which apply the control schemes. The number of controlled UDP applications at routers R0 and R2 increases from 5 to 36, to 72 and to 144 for each experimental set. Table III shows the variation of different aspects of the topology in order to conduct these sets of experiment.

The second circumstance of the scalability is to increase the percentage of controlled UDP flows in total traffic. This is achieved by increasing the number of controlled UDP nodes at two ends of the topology while keeping the other parameters unchanged. There are also four experimental sets in this case. For each set, the number of controlled UDP nodes increases from 5 to 20, to 50 and to 80 while the ratio of TCP flows to total traffic decreases correspondingly.

Table III. Details of topology #2 to see the effect of increasing percentage of controlled UDP flows in the network traffic while keeping the overall contribution of UDP flows to the total traffic constant.

Experiment	1	2	3	4
No. of Controlled UDP Flows (R0-R2)	**5**	**36**	**72**	**144**
No. of Controlled UDP Flows (R2-R0)	**5**	**36**	**72**	**144**
Max. Bit-Rate / Controlled UDP Flow (kbps)	96	96	96	96
Max. Total Bit-Rate of Controlled UDP Flow (kbps)	960	6912	13824	27648
Min. Bit-Rate / Controlled UDP Flow (kbps)	36	36	36	36
Min. Total Bit-Rate of Controlled UDP Flow (kbps)	360	2592	5184	10368
No. of UDP Nodes (R0-R1)	5	5	5	5
Size of Pkts. (Bytes)	512	512	512	512
Inter-Dept. Time of Pkts. (sec.)	0.002	0.002	0.002	0.002
Total Bit-Rate (kbps)	10240	10240	10240	10240
No. of UDP Nodes (R1-R0)	5	5	5	5
Size of Pkts. (Bytes)	512	512	512	512
Inter-Dept. Time of Pkts. (sec.)	0.002	0.002	0.002	0.002
Total Bit-Rate (kbps)	10240	10240	10240	10240
No. of UDP Nodes (R0-R2)	20	20	20	20
Mean Bit-Rate / Flow (kbps)	**800**	**700**	**600**	**400**
Total Bit-Rate (kbps)	16000	14000	12000	8000
No. of UDP Nodes (R2-R0)	20	20	20	20
Mean Bit-Rate / Flow (kbps)	**820**	**720**	**620**	**420**
Total Bit-Rate (kbps)	16400	14400	12400	8400
Max. of (with bursty UDP flows)	1.78	12.39	23.55	42.85
Min. of (with bursty UDP flows)	0.67	5.04	10.36	21.95
Max. of (without bursty UDP flows)	2.87	19.57	36.17	62.77
Min. of (without bursty UDP flows)	1.09	8.36	15.53	38.74

CHAPTER III

LINEAR MODEL FOR SYSTEM IDENTIFICATION

In system identification, modeling techniques can be divided into three of the following categories:

(1) White box models: white box modeling technique interprets systems' dynamics in term of mathematical equations, usually a system of PDE or ODE. These equations are derived from physical, chemical, etc. laws. This type of modeling is the most preferred one because it gives a good extrapolation of the system's responses and is highly reliable. The drawback of this method is time consuming and a well understanding of the processes.

(2) Black box models: differing to white box models which require of expertise knowledge, black box models rely mainly upon measured data of the interested system. The model's structure and parameters can be evaluated from these data. Black box techniques are very powerful in modeling complex processes where there is little or no domain expertise. However, it has unreliable extrapolations of the system's responses and provides little understanding of the underlying physical processes.

(3) Grey box models: As its name inferring, grey box models are the combination between white and black box models. Particularly, model's structures are usually chosen by expert knowledge while the determinations of model's parameters are mainly relied on measured data.

It is impossible to mathematically describe a network topology in term of white box modeling. The reasonable approach is through measured data using black box techniques. Specifically, the linear system identification methods are at interest. Several well-known linear models are Auto-regressive, Auto-regressive Exogenous, Auto-regressive Moving Average and Auto-regressive Moving Average with Exogenous terms. These models describe the system with a number of weighted signals including: input, output and white noise. The weighting parameters are determined by optimizing methods such as linear regression. The advantages of these linear models are computationally cheap development and simple processes in designing MPC controllers. Figure 3.1 shows the general linear model structure. Four common forms mentioned above can be derived from this general model.

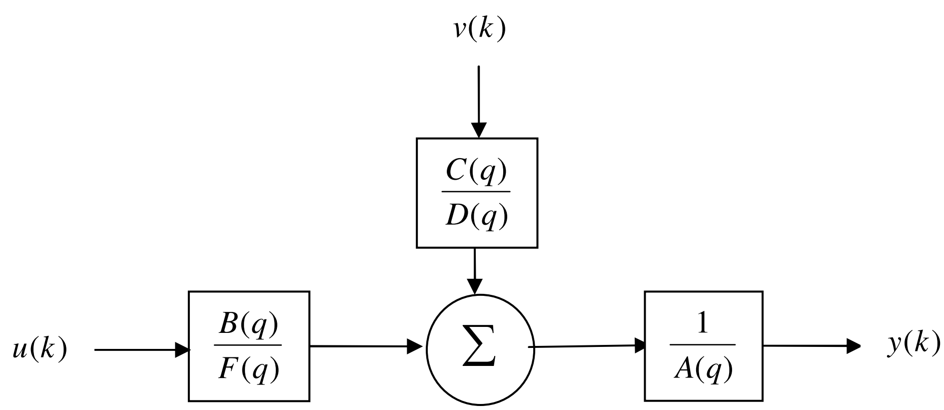

Fig. 3.1. General linear model structure.

Various documents describe linear system identification techniques can be found at different resources. For example, Kommaraju [14] provides a very good summary of the linear system identification models. Within this research, only the Auto-regressive Exogenous model (ARX) is at particular interest.

A. Auto-regressive Exogenous Model

ARX is the simplest and the most used model structure in linear system identification. The general SISO ARX model can be expressed by the following linear difference equation:

$$y(k) = a_1 y(k-1) + \ldots + a_{n_y} y(k-n_y) + b_1 u(k-n_k) + \ldots + b_{n_u} u(k-n_u-n_k+1) + v(k), \tag{3.1}$$

where $u(k)$ and $y(k)$ are the input and the output of the SISO ARX model, $v(k)$ is the noise disturbance, n_y and n_u are the number of past outputs and the number of past inputs used in the model, and n_k is the pure time delay (i.e.the dead time) in the system. The coefficients $a_1,\ldots, a_{n_y}$ and $b_1, \ldots, b_{n_u}$ are known as the model parameters. The noise disturbance $v(k)$ is usually assumed to be white noise. Figure 3.2 depicts the structure of ARX models.

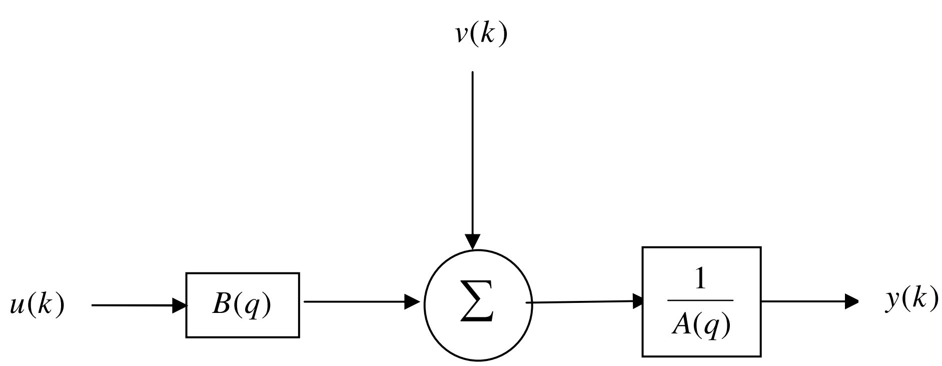

Fig. 3.2. ARX model structure.

Based on ARX model of equation (3.1), the following Single Step Predictor of the system output can be obtained:

$$\hat{y}(k|k-1,\theta)=\phi(k)\theta, \tag{3.2}$$

where:

$$\phi=[y(k-1),\ldots,y(k-n_y),u(k-n_k),\ldots,u(t-n_u-n_k+1)],$$
$$\theta=[a_1,\ldots,a_{n_y},b_1,\ldots,b_{n_u}]^T.$$

Equation (3.2) is in the form of a linear regression with the model parameter vector θ knowing as regression vector. The parameter vector θ is estimated using the least-squares method to minimize the mean-square of the prediction error.

The Auto-regressive (AR) model is a special case of the ARX model where only past values of the output are used for modeling the system. The AR model is a time-series model, also known as a Box-Jenkins model.

Here, Matlab's System Identification Toolbox is used to estimate the model parameters.

B. Designed ARX Models

The contribution of this research is to improve Bhattacharya's MPC strategy by introducing a feed-forward term in addition to feedback terms. To compare the proposed MPC strategy with Bhattacharya's MPC scheme, two different MPC controllers are constructed from two different ARX models:

The first one is the ARX model designed by Bhattacharya [9] without the feed-forward term. Specific information of the model are ARX [14, 15, 1] with n_y = 14, n_u = 15, n_k = 1. The denominator $A(q)$ and numerator $B(q)$ of the model respectively are:

$$\begin{aligned} A(q) = 1 - 0.1519q^{-1} - 0.5841q^{-2} + 0.2529q^{-3} + 0.04036q^{-4} - \\ 0.1455q^{-5} - 0.121q^{-6} - 0.189q^{-7} - 0.06726q^{-8} + 0.07563q^{-9} + \\ 0.2044q^{-10} + 0.08776q^{-11} - 0.1296q^{-12} + 0.2397q^{-13} - 0.1537q^{-14}, \end{aligned} \tag{3.5}$$

$$\begin{aligned} B(q) = 7.691q^{-1} - 3.891q^{-2} - 3.62q^{-3} + 5.121q^{-4} - 2.577q^{-5} - \\ 0.1723q^{-6} - 2.662q^{-7} - 2.887q^{-8} + 0.5515q^{-9} + 2.167q^{-10} + \\ 3.374q^{-11} - 0.8988q^{-12} + 2.346q^{-13} + 0.9643q^{-14} - 1.048q^{-15}. \end{aligned} \tag{3.6}$$

The second ARX model contains information of the current input or feed-forward term. The proposed ARX model has the structure of ARX [14, 15, 0] with n_y = 14, n_u = 15, n_k = 0. The denominator $A(q)$ and numerator $B(q)$ of the model respectively are:

$$\begin{aligned} A(q) = 1 - 0.6854q^{-1} + 0.05831q^{-2} + 0.1587q^{-3} - 0.2191q^{-4} + \\ 0.047715q^{-5} - 0.4014q^{-6} + 0.2593q^{-7} + 0.007027q^{-8} - 0.02644q^{-9} + \\ 0.1925q^{-10} - 0.1297q^{-11} + 0.03742q^{-12} - 0.03059q^{-13} + 0.07877q^{-14}, \end{aligned} \tag{3.7}$$

$$\begin{aligned} B(q) = 6.279 + 1.414q^{-1} - 3.701q^{-2} + 1.257q^{-3} - 0.5105q^{-4} - 1.119q^{-5} - \\ 2.081q^{-6} - 0.7927q^{-7} + 1.477q^{-8} - 0.04757q^{-9} + 0.9979q^{-10} + \\ 0.2923q^{-11} - 0.431q^{-12} + 0.1957q^{-13} + 0.4585q^{-14} + 0.6146q^{-15}. \end{aligned} \tag{3.8}$$

C. Measuring Metrics for Designed ARX Models

The performance of the two models is measured in terms of four different metrics:

(1) Mean Square Error (MSE): MSE is the ratio between the sum of the square of the prediction error and the sum of the square of the input data. MSE is defined by:

$$MSE = \frac{\sum_{j=1}^{N_S}(A(j)-\widehat{A}(j))^2}{\sum_{j=1}^{N_S}A(j)^2}\times 100, \tag{3.9}$$

where, N_S is the total number of samples in the output data (accumulation), $A(j)$ is the jth sample of the actual accumulation, and $\hat{A}(j)$ is the one step head prediction of the jth sample of the accumulation signal. MSE gives an idea of the overall quality of the prediction. The lower the MSE is the better is the performance of the predictor.

(2) Maximum Absolute Error (MAE): MAE is the maximum error between the actual accumulation signal and the one step ahead prediction of the accumulation signal. It is defined as:

$$MAE = \max_{1\le j\le N_S} |A(j)-\widehat{A}(j)|. \tag{3.10}$$

This definition of error tries to provide some insight into the worst case error. The lower the MAE the better is the performance of the predictor.

(3) Fit: The norm of a vector is a scalar that gives some measure of the magnitude of its elements. The fit of a vector is defined as:

$$\| A \|_2 = \left(\sum_{j=1}^{N_S} |A(j)|^2\right)^{1/2}, \tag{3.11}$$

$$fit = \left(\frac{1-\|\widehat{A}-A\|_2}{\|A-\overline{A}\|_2}\right)\times 100, \tag{3.12}$$

where, $\hat{A}$ represents the vector of the SSP of the accumulation signal, A is the vector of the measured values of the accumulation signal, and A is vector of the mean value of the measured accumulation signal from the simulation experiments. The higher the fit is the better is the performance of the predictor.

(4) Probability of Normalized Error to be less than 10 percent (PNE10): the previous three metrics are not providing enough insight into the details of the prediction error. Indeed, although the previous three metrics are pointing towards good prediction results, yet the predictions might not as good as expected. Bhattacharya [9] suggested a new metric to judge the quality of predictions. This metric shows the probability of the normalized absolute prediction error to be less than 10 percent.

$$PNE10 = P\left(\frac{|\hat{A}(j) - A(j)|}{A(j)} \leq 0.1\right), \tag{3.13}$$

such that $j \in \{1, \ldots, N_S\}$. In this formula, the higher PNE10 is the better the prediction.

The performance results of two models in terms of the above measurement metrics are shown in Table IV.

Table IV. One-step-ahead predictions of the ARX models in terms of defined metrics.

Percent Loss	MSE (%)	MAE (Bytes)	Fit (%)	PNE10
ARX [14, 15, 1]				
3	2.44	2215.0	24.76	0.52
5	2.50	2342.5	24.75	0.50
9	2.84	2302.5	19.52	0.50
15	3.52	2282.5	9.66	0.47
ARX [14, 15, 0]				
3	0.475	655.5	72..98	0.854
5	0.53	694	71.93	0.836
9	0.84	900.3	67.52	0.778
15	0.82	823.2	67.05	0.772

The table shows that the single step prediction of the accumulation signal is much better with the introduction of feed-forward input term.

Figure 3.3 shows the single step prediction of mean accumulation and the prediction error of the ARX designed by Bhattacharya. The mean accumulation prediction and error of the proposed ARX is shown in Figure 3.4.

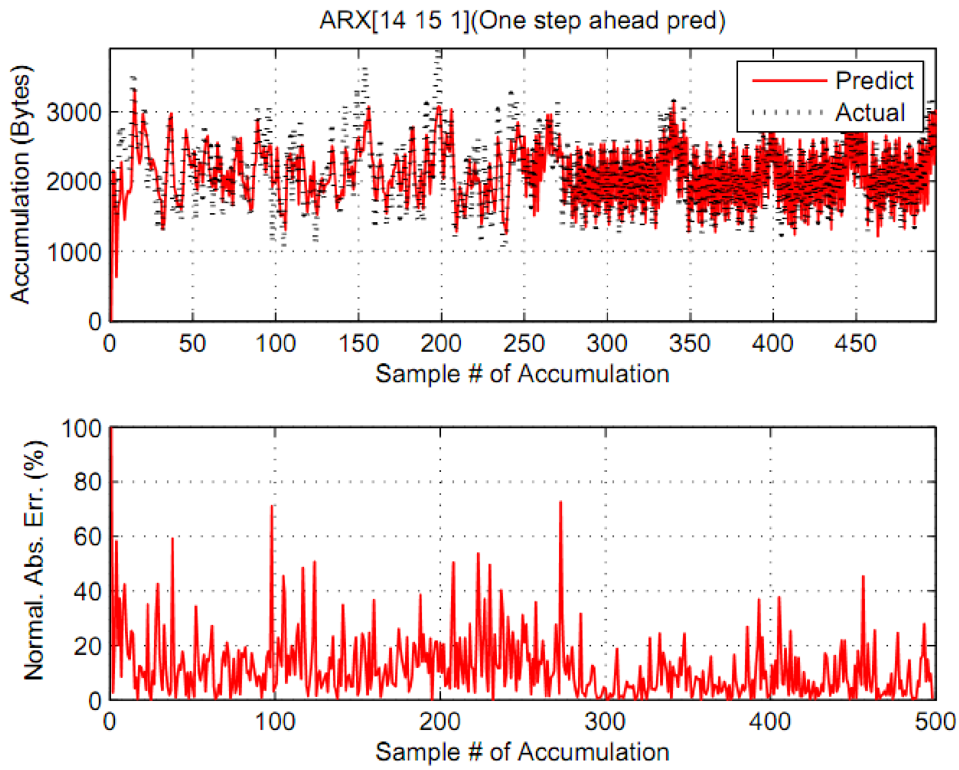

Fig. 3.3. One-step-ahead prediction of accumulation signal in 3% CLR network using ARX model designed by Bhattacharya[9].

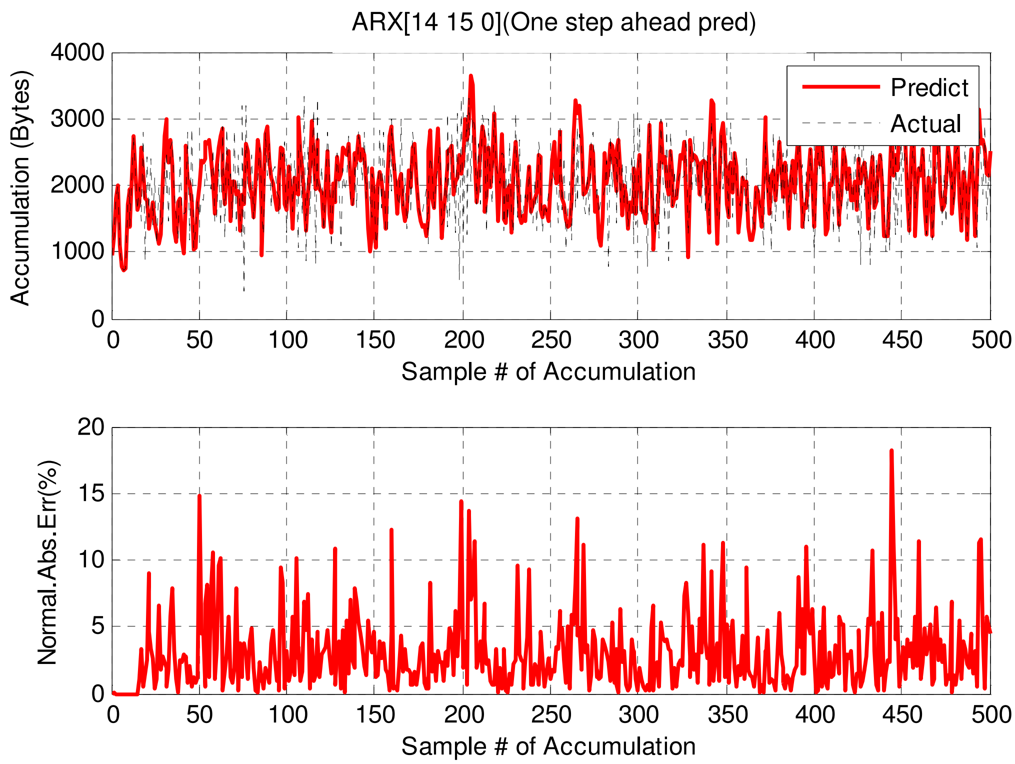

Fig. 3.4. One-step-ahead estimation of accumulation signal in 3% CLR network using the proposed ARX model.

CHAPTER IV

MODEL PREDICTIVE CONTROL

A. Model Predictive Control Law

One of the features that make TCP dominates the Internet is its congestion control scheme. However, this is a simple reactive control strategy which tries to solve the impact after congestion occurs rather than to prevent it. This is not an effective way of doing control. An ideal congestion controller should be able to prevent congestion to occur by forecasting its future status. This type of control strategy is well-known as predictive control.

One of the most famous predictive control strategies is Model Predictive Control (MPC). MPC is an advanced method of process control. It relies on dynamic models of the processes to calculate the control input through minimizing an objective function. It is very well established for linear models such as ARX and ARMAX. Control principles of MPC can be summarized as follow:

- A series of future control inputs are calculated by predicting future outputs using the system model and minimizing a specified objective function.
- Only the first control input of the calculated sequence is applied to the process at the next time instant.

Specifically, the MPC is implemented in three steps:

(1) The future outputs for a certain prediction horizon N are predicted at instant k using the process model. The predicted outputs $\hat{y}(k+N|k)$ depend on the known values of past inputs and outputs up to instant k, as well as on the future control inputs up to $u(k+N|k)$ and predicted outputs up to $\hat{y}(k+N-1|k)$.

(2) The set of future control signals is calculated by optimizing an objective function in order to keep the process tracking a reference trajectory $r(k + N)$. The objective function is a quadratic function of the errors between the predicted output signal and the reference trajectory. The control effort is included in the objective function in most cases.

(3) Among all of the calculated control signals, $u(k)$ is the only one that is applied to the process. All the others are rejected. At the next sampling instant, step 1 and step 2 are repeated with new measured values of input and output signals.

Figure 4.1 shows the schematic of MPC strategy. In this figure, u represents sending bit rate of real-time conferencing applications and y represents accumulation signal which is directly related with the network congestion.

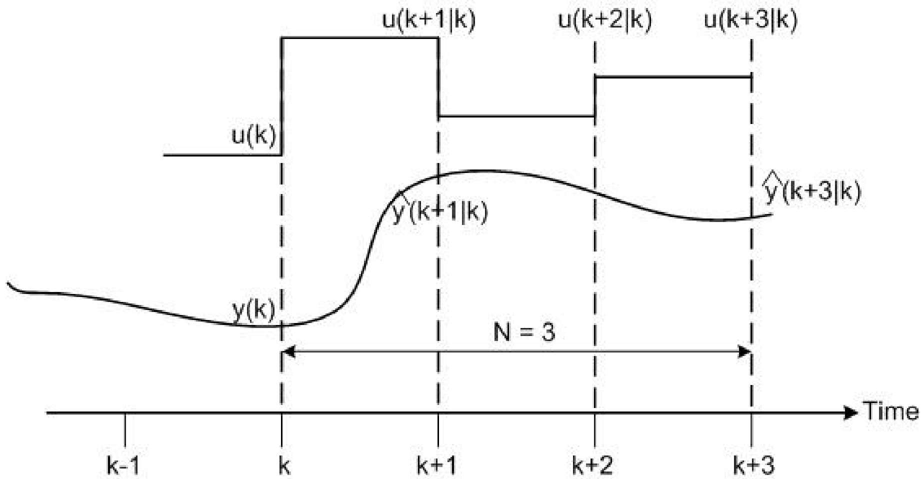

Fig. 4.1. Schematic of MPC strategy.

B. Formulations of the Model Predictive Control Strategies

In this research, two MPC strategies are considered. The first one follows Bhattacharya's proposed scheme. Specifically, it is developed based on the ARX [14, 15, 1]. This model does not take the current input into account; therefore, the prediction results are not highly accurate. Consequently, the controller shows erratic responses. Detailed formulation of the first MPC can be found in Bhattacharya [9] or [10].

In the second MPC scheme, it is proposed to introduce the feed-forward term into the ARX model. In particular, the proposed MPC controller is constructed based upon the ARX [14, 15, 0]. This leads to a small change in the formulation but the developing process remains the same:

The linear difference equation describing an end-to-end single flow is:

$$\begin{aligned} y(k) &= \alpha_1 y(k-1) + \alpha_2 y(k-2) + \cdots + \alpha_n y(k-n) + \\ & b_0 u(k) + b_1 u(k-1) + +b_2 u(k-2) + \cdots + b_m u(k-m) + \frac{d(k)}{\Delta}. \end{aligned} \tag{4.1}$$

By multiplying both sides with the operator $\Delta = 1 - q^{-1}$. We get:

$$y(k) = \alpha_1' y(k-1) + \alpha_2' y(k-2) + \cdots + \alpha_n' y(k-n) + \alpha_{n+1}' y(k-n-1) + \\ b_0 \Delta u(k) + b_1 \Delta u(k-1) + + b_2 \Delta u(k-2) + \cdots + b_m \Delta u(k-m) + d(k). \tag{4.2}$$

Assuming the current accumulation $y(k)$ and desired future input to the network are known and let $a_i = -\alpha_i'$, the simulator form of (3.19) is:

$$\hat{y}(k+1 | k-1) + a_1 y(k) + a_2 y(k-1) + \ldots + a_n y(k-n+1) + a_{n+1} y(k-n) = \\ b_0 \Delta u(k+1) + b_1 \Delta u(k) + b_2 \Delta u(k-1) + \ldots + b_m \Delta u(k-m+1). \tag{4.3}$$

The equations for the next three steps of the prediction are:

$$\hat{y}(k+2 | k-1) + a_1 \hat{y}(k+1 | k-1) + a_2 y(k) + \ldots + a_{n+1} y(k-n+1) = \\ b_0 \Delta u(k+2) + b_1 \Delta u(k+1) + b_2 \Delta u(k) + \ldots + b_m \Delta u(k-m+2), \tag{4.4}$$

$$\hat{y}(k+3 | k-1) + a_1 \hat{y}(k+2 | k-1) + a_2 \hat{y}(k+1 | k-1) + \ldots + a_{n+1} y(k-n+2) = \\ b_0 \Delta u(k+3) + b_1 \Delta u(k+2) + b_2 \Delta u(k+1) + \ldots + b_m \Delta u(k-m+3), \tag{4.5}$$

$$\hat{y}(k+4 | k-1) + a_1 \hat{y}(k+3 | k-1) + a_2 \hat{y}(k+2 | k-1) + \ldots + a_{n+1} y(k-n+3) = \\ b_0 \Delta u(k+4) + b_1 \Delta u(k+3) + b_2 \Delta u(k+2) + \ldots + b_m \Delta u(k-m+4). \tag{4.6}$$

The above equations can be written in the following matrix form:

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ a_1 & 1 & 0 & 0 \\ a_2 & a_1 & 1 & 0 \\ a_3 & a_2 & a_1 & 1 \end{pmatrix}}_{C_A} \underbrace{\begin{pmatrix} \hat{y}(k+1 | k-1) \\ \hat{y}(k+2 | k-1) \\ \hat{y}(k+3 | k-1) \\ \hat{y}(k+4 | k-1) \end{pmatrix}}_{\hat{y}} + \underbrace{\begin{pmatrix} a_1 & a_2 & \ldots & a_n & a_{n+1} \\ a_2 & a_3 & \ldots & a_{n+1} & 0 \\ a_3 & a_4 & \ldots & 0 & 0 \\ a_4 & a_5 & \ldots & 0 & 0 \end{pmatrix}}_{H_A} \underbrace{\begin{pmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-n) \end{pmatrix}}_{y} =$$

$$\underbrace{\begin{pmatrix} b_0 & 0 & 0 & 0 \\ b_1 & b_0 & 0 & 0 \\ b_2 & b_1 & b_0 & 0 \\ b_3 & b_2 & b_1 & b_0 \end{pmatrix}}_{C_B} \underbrace{\begin{pmatrix} \Delta u(k+1) \\ \Delta u(k+2) \\ \Delta u(k+3) \\ \Delta u(k+4) \end{pmatrix}}_{\Delta u_F} + \underbrace{\begin{pmatrix} b_1 & b_2 & \ldots & b_{m-1} & b_m \\ b_2 & b_3 & \ldots & b_m & 0 \\ b_3 & b_4 & \ldots & 0 & 0 \\ b_4 & b_5 & \ldots & 0 & 0 \end{pmatrix}}_{H_B} \underbrace{\begin{pmatrix} \Delta u(k) \\ \Delta u(k-1) \\ \vdots \\ \Delta u(k-m+1) \end{pmatrix}}_{\Delta u}. \tag{4.7}$$

The shorter version of equation 3.24 is:

$$
\begin{aligned}
&C_A \hat{y} = -H_A y + C_B \Delta u_F + H_B \Delta u \\
&\Rightarrow \hat{y} = -C_A^{-1} H_A y + C_A^{-1} C_B \Delta u_F + C_A^{-1} H_B \Delta u \; .
\end{aligned} \tag{4.8}
$$

Let $H = C_A^{-1} C_B$, $P = C_A^{-1} H_B$, and $Q = -C_A^{-1} H_A$. Therefore, the predicted future values of $\hat{y}(k|k-1)$, $\hat{y}(k+1|k-1)$,$\hat{y}(k+2|k-1)$, $\hat{y}(k+3|k-1)$ are provided by:

$$
\hat{y} = Qy + H \Delta u_F + P \Delta u \, . \tag{4.9}
$$

The MPC control strategy is devised by minimizing the objective function *J* defined by the following expression w.r.t. the future control inputs Δu_F:

$$
\begin{aligned}
&J_{\Delta u_F} = \| e \|_2^2 + \lambda \| \Delta u_F \|_2^2 \\
&\Rightarrow J_{\Delta u_F} = \| r - \hat{y} \|_2^2 + \lambda \| \Delta u_F \|_2^2 \, ,
\end{aligned} \tag{4.10}
$$

where, *r* represents the value of the reference signal in the next few sampling instants.

Substitute $\hat{y}$ from (3.9) to (3.10) give:

$$
J_{\Delta u_F} = \| r - Qy - H \Delta u_F - P \Delta u \|_2^2 + \lambda \| \Delta u_F \|_2^2 \tag{4.11}
$$

$$
\Leftrightarrow J_{\Delta u_F} = (r - Qy - H \Delta u_F - P \Delta u)^T (r - Qy - H \Delta u_F - P \Delta u) + \lambda \Delta u_F^T \Delta u_F \tag{4.12}
$$

$$
\Leftrightarrow J_{\Delta u_F} = 2 \Delta u_F^T (-H^T r + H^T P \Delta u + H^T Q y) + \Delta u_F^T (H^T H + \lambda I) \Delta u_F + \chi \, , \tag{4.13}
$$

where, χ is the sum of terms that do not contain $\Delta \underline{u}_F$. Minimize equation (3.13) w.r.t. $\Delta \underline{u}_F$ give:

$$
H^T (-r + P \Delta u + Qy) + (H^T H + \lambda I) \Delta u_F = 0. \tag{4.14}
$$

From equation (3.14), the control law is:

$$\Delta u_F = (H^T H + \lambda I)^{-1} H^T (r - P\Delta u - Qy), \tag{4.15}$$

with $\Delta u_F = [\Delta u(k+1), \Delta u(k+2), \Delta u(k+3), \Delta u(k+4)]^T$. Only $\Delta u(k+1)$ is at interest:

$$u(k+1) = \Delta u(k+1) + u(k)\ . \tag{4.16}$$

Some notations can be drawn from Bhattacharya's work [9] as follow:

- In equations (3.15), there are two parameters that need to be decided.
    - λ: responsible for penalizing the change of input (bit rate) in the objective functions. The higher λ is, the costlier is the input change and vice versa.
    - *r*: control reference vector is kept constant. If the reference level is too high, the network remains congested all the time. In contrast, if the reference level is too low, the control law becomes too conservative with the low bit rates dominate during a session. Both of the cases may result in correspondingly low QoS. In this research, λ is determined to be 10 for both MPC schemes. The accumulation reference was chosen to be neutral at 1825.
- The MPC formulation has been done without concerning the limitation of the encoder. That is, there are only six possible values for $u(k+1)$ instead of unlimited range. To prevent the input signals from violating their bounds, quantization port is settle according to the following equation:

$$u(k+1)=\begin{cases} 90, \text{if } u(k+1)<105 \\ 120, \text{if } 105 \le u(k+1)<135 \\ 150, \text{if } 135 \le u(k+1)<165 \\ 180, \text{if } 165 \le u(k+1)<195 \\ 210, \text{if } 195 \le u(k+1)<225 \\ 240, \text{if } u(k+1) \ge 225 \end{cases}. \tag{4.17}$$

- Since the bounds of the encoder were not considered in the objective function, the formulated MPC strategy is not optimal. Explicit handling of the constraints requires the solution of a Quadratic Programming (QP) problem, an optimization problem with a quadratic objective function and linear constraints. This usually takes a lot of computing power and time. Therefore, a flow control scheme, based on solving a QP problem at each time step, is difficult to implement in practice.

CHAPTER V

EXPERIMENTAL RESULTS

A. Validation of the Control Strategies

On simulation topology #1, the performance of two MPC controllers is validated by six experimental scenarios corresponding to different comprehensive loss rate of 3%, 5%, 7%, 9% and 15%. The loss rates were created by varying bandwidth capacity of the backbone between routers R0 and R1. Table I shows R0-R1 bandwidth capacities together with average comprehensive loss rates.

Figures 5.1-5.5 are performances on 3% comprehensive loss rate scenario. Here, the results are compared among the flows controlled by MPC strategies, uncontrolled flows with highest bit rate (HBR) and lowest bit rate (LBR). In the case of HBR, real-time multimedia applications constantly send out the highest bit rate of 96 (Kbps). QoS of such applications are usually high but they consume a large amount of backbone's bandwidth. Therefore, HBR is sensitive to network congestion. On the other hand, congestion is less likely a problem in LBR in which the interested flows use lowest bandwidth of 36 (Kbps). As the result, LBR usually does not have a good QoS. In Figure 5.1, the quality of audio is expressed in term of mean MOS. Here, MOS is calculated after every second and averaged on the total of 120s running time. As can be seen, multimedia flows controlled by MPC schemes have MOS as good as HBR and better than LBR. The advantage of MPC strategies reflects on their ability to save bandwidth.

In fact, Figure 5.2 indicates that both MPC models use much less bandwidth than HBR in order to achieve a similar QoS.

According to Figures 5.1 and 5.2, the proposed MPC scheme does not show any clear benefit over Bhattacharya's strategy. However, it helps to eliminate the deficiency of the model predictive controller proposed by Bhattacharya. As in Figures 5.3 and 5.4, the controller following Bhattacharya's strategy show erratic response of which the bandwidth jumps frequently between lowest and highest values, which is not desirable. In contrast, the proposed controller has a more stable response and its bandwidth varies around its mean value of 59.59 Kbps.

To measure the effect of bandwidth saving, the probability P_{saving} is introduced. Taking into account that a controlled flow does not free any of the backbone's bandwidth when using its highest bit rate, P_{saving} is calculated as follow:

$$P_{saving} = 1 - \frac{N_{HBR}}{N_{total}} ,$$

where: $N_{HBR}$is the number of samples at which the controllers use highest bit rate. N_{total} is the total number of sample points. Here, each sample is collected every 240ms during 120s of the experiments; this comprises 500 samples for N_{total}.

From Figure 5.5, the probability of bandwidth saving is about 99% for the proposed controller. This means that the saving effect almost always occurs during the execution of the proposed model. In the case of MPC strategy proposed by Bhattacharya, P_{saving} is only about 58%.

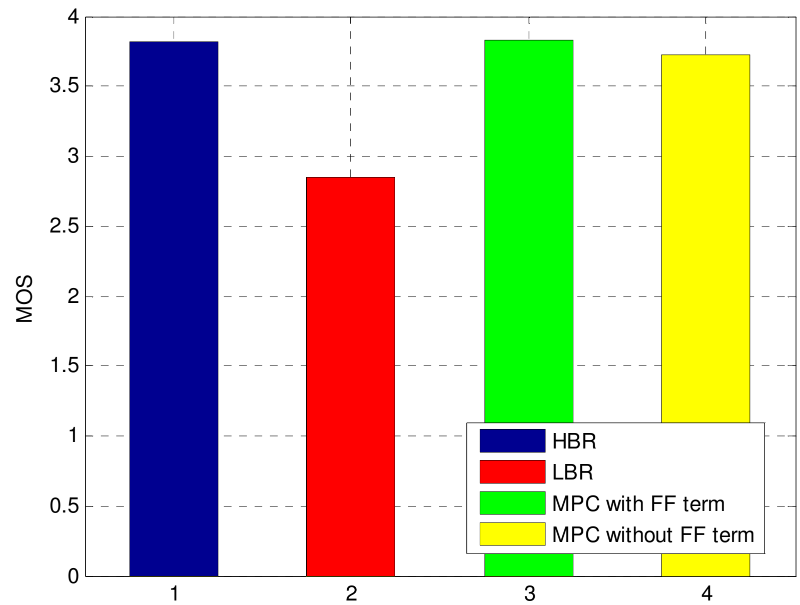

Fig. 5.1. Performance of MPC controllers in term of mean MOS of five controlled UDP flows compared to those of uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in network topology 1 with 3% CLR.

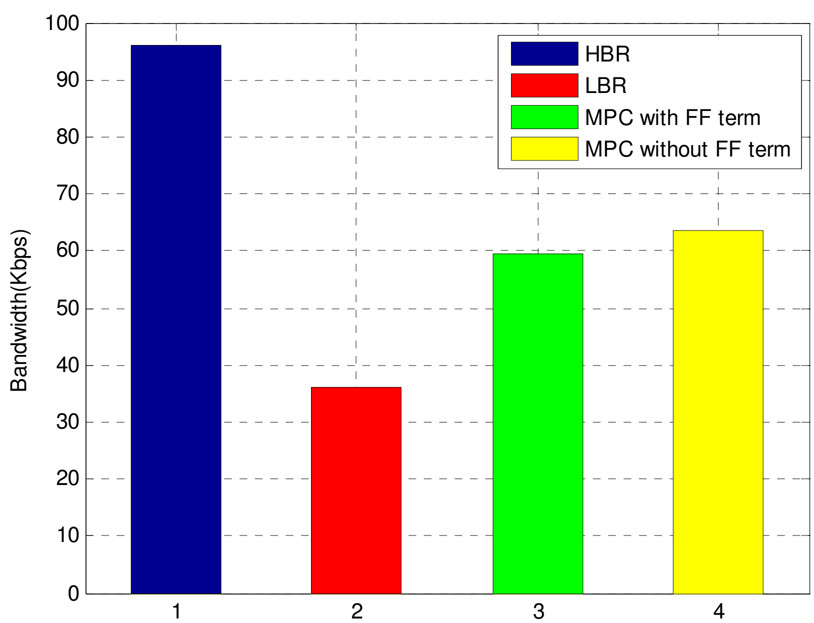

Fig. 5.2. Bandwidth used by controlled UDP flows compared to uncontrolled flows with highest (HBR) and lowest (LBR) bit rate (3% CLR).

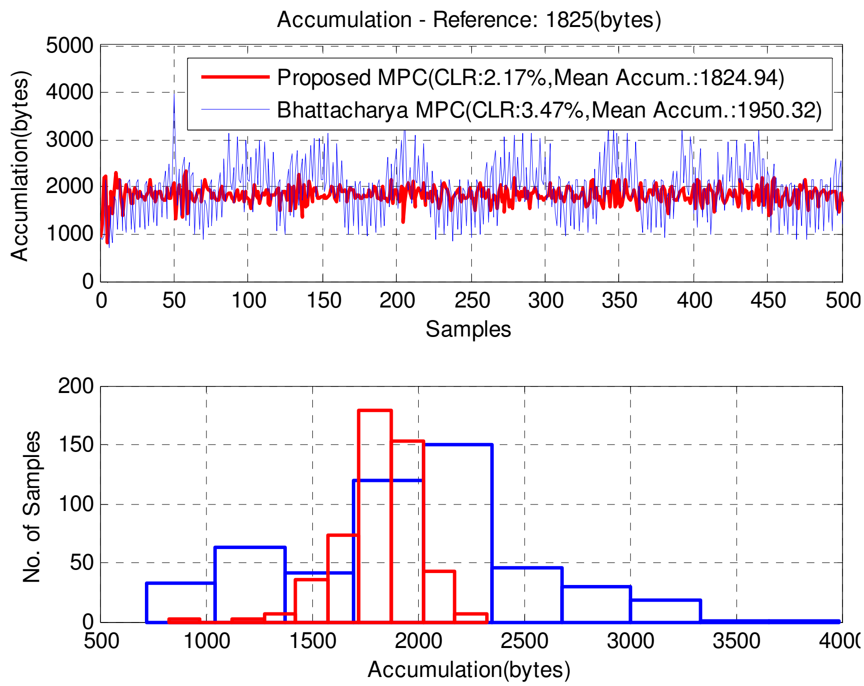

Fig. 5.3. Accumulation responses and their histogram of two MPC controllers (3% CLR).

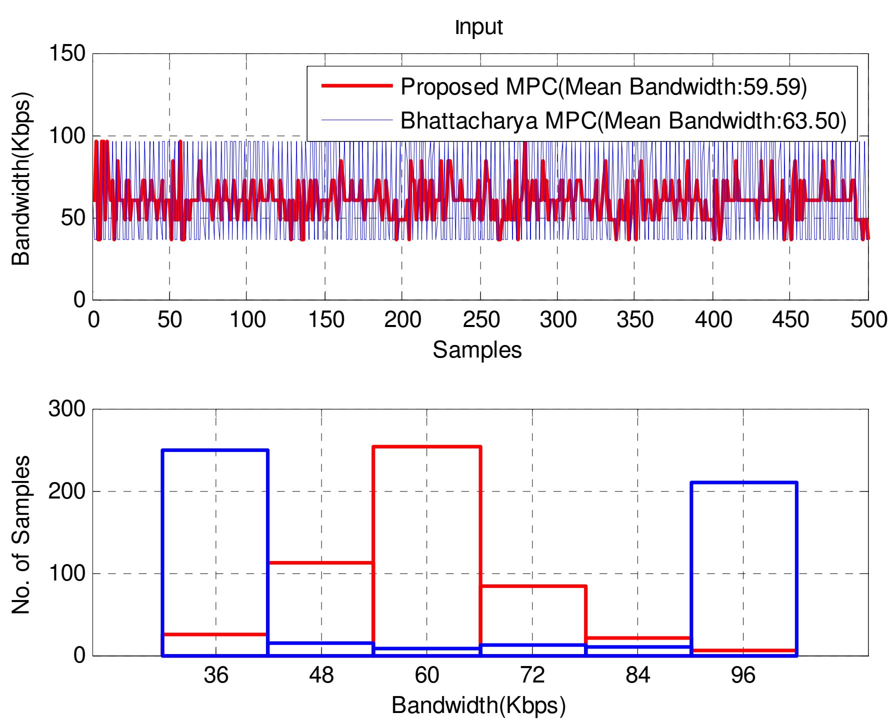

Fig. 5.4. Control inputs and their histogram of two MPC controllers (3% CLR).

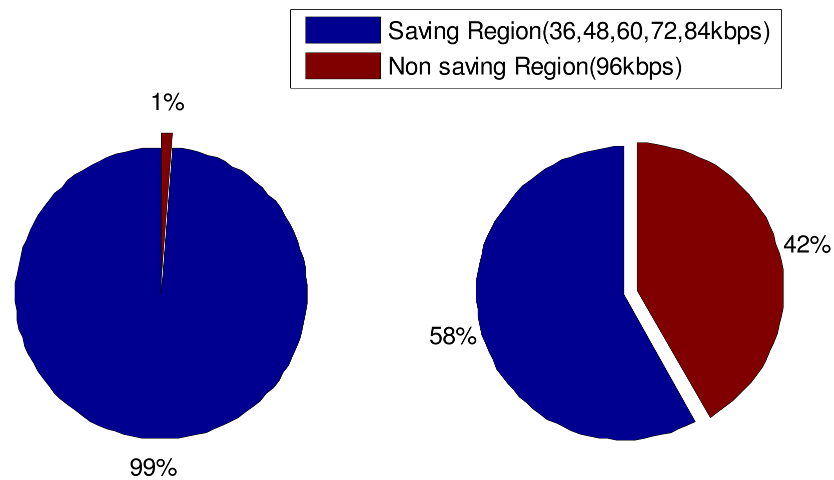

Fig. 5.5. Probability of bandwidth savings: proposed control scheme (left) and Bhattacharya scheme (right) (3% CLR).

Figures 5.6-5.10 are performances in the cases of 5% CLR. Similar to the previous case, these figures show that both controllers can gain QoS as good as those of HBR while use less bandwidth and the proposed controller is more desirable with P_{saving} remains high around 99% compared to 58% of Bhattacharya's MPC scheme.

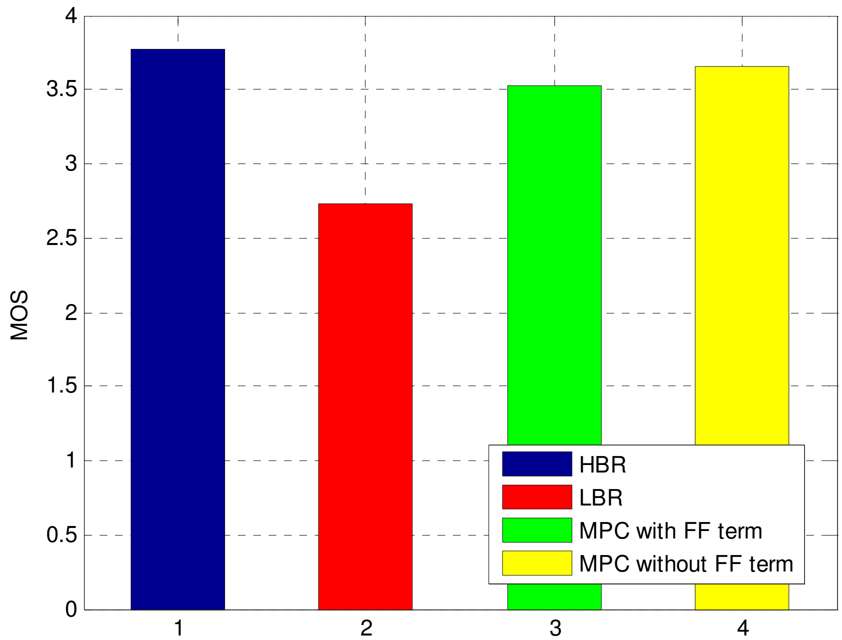

Fig. 5.6. Performance of MPC controllers in term of mean MOS of five controlled UDP flows compared to those of uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in network topology 1 with 5% CLR.

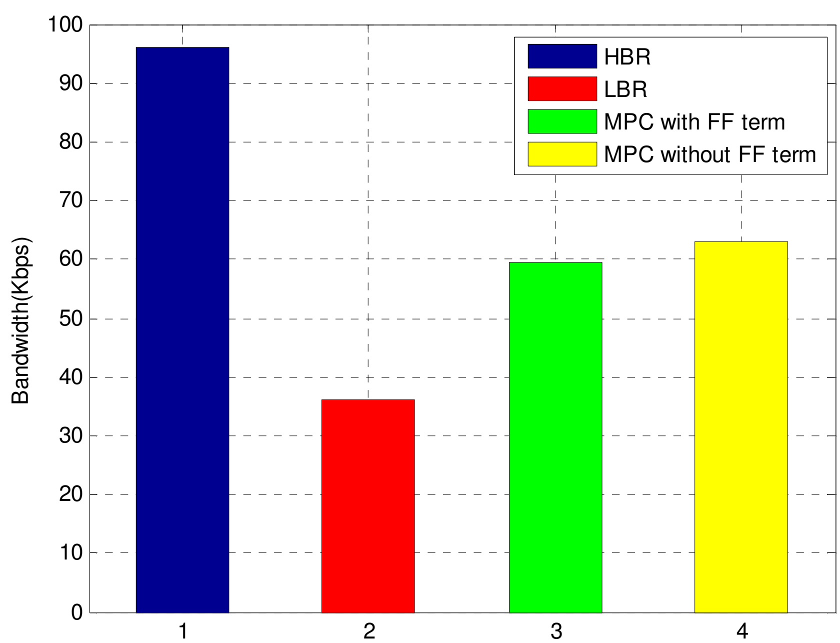

Fig. 5.7. Bandwidth used by controlled UDP flows compared to uncontrolled flows with highest (HBR) and lowest (LBR) bit rate (5% CLR).

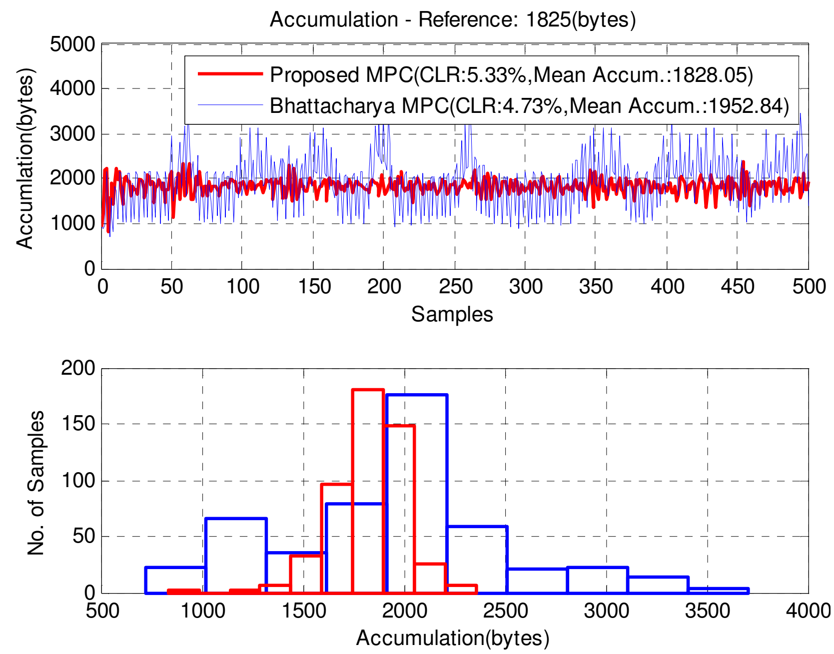

Fig. 5.8. Accumulation responses and their histogram of two MPC controllers (5% CLR).

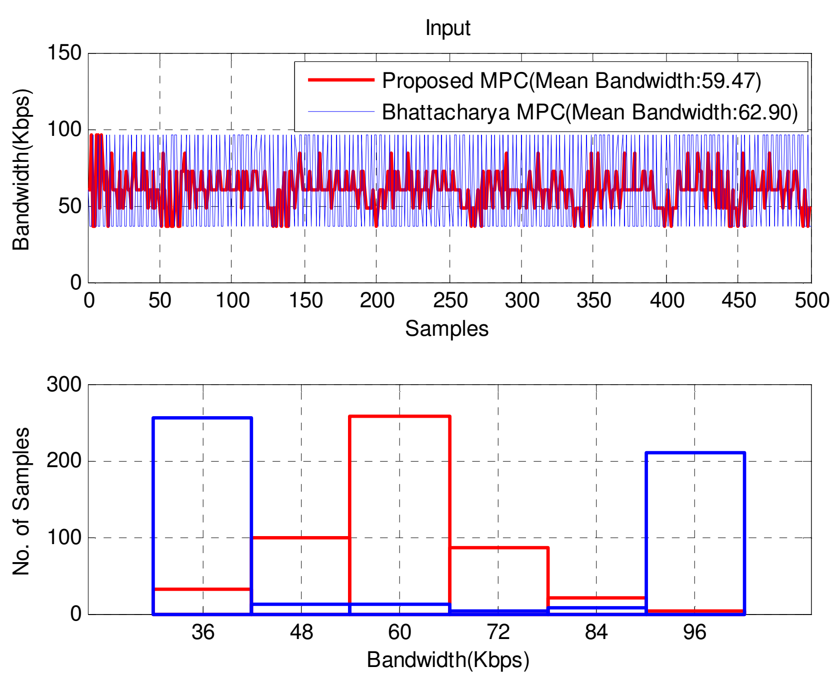

Fig. 5.9. Control inputs and their histogram of two MPC controllers (5% CLR).

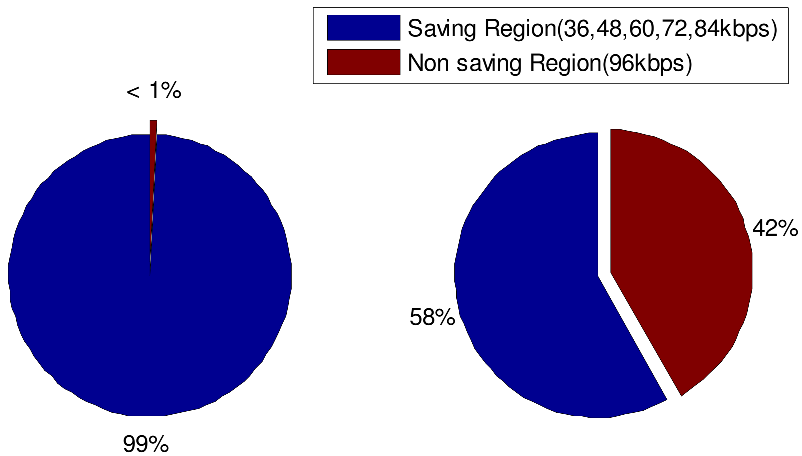

Fig. 5.10. Probability of bandwidth savings: proposed control scheme (left) and Bhattacharya scheme (right) (5% CLR).

Figures 5.11-5.15 are performances in the cases of 7% CLR. Again, both controllers can still gain good QoS while save a considerable amount of bandwidth. The modified controller has more stable response and higher probability of bandwidth savings compared to one proposed by Bhattacharya.

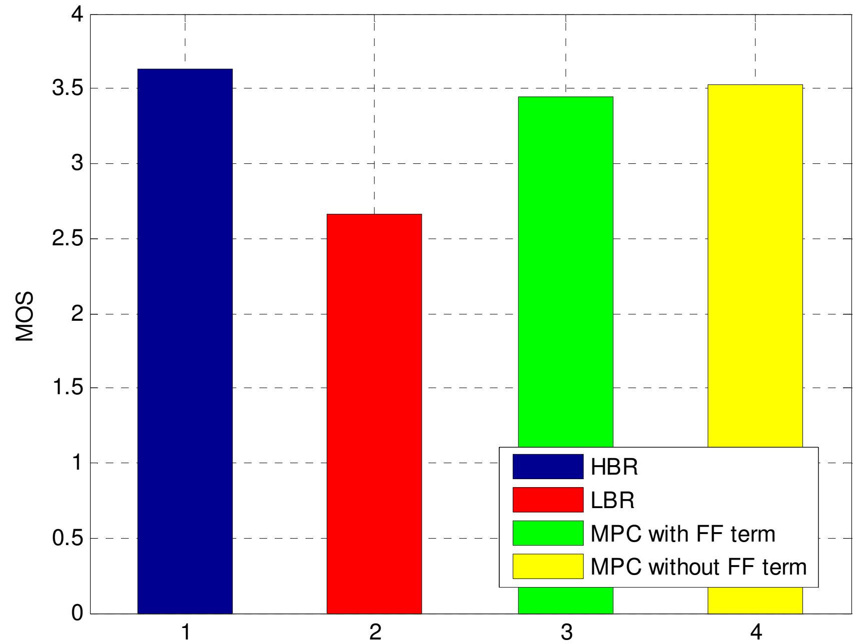

Fig. 5.11. Performance of MPC controllers in term of mean MOS of five controlled UDP flows compared to those of uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in network topology 1 with 7% CLR.

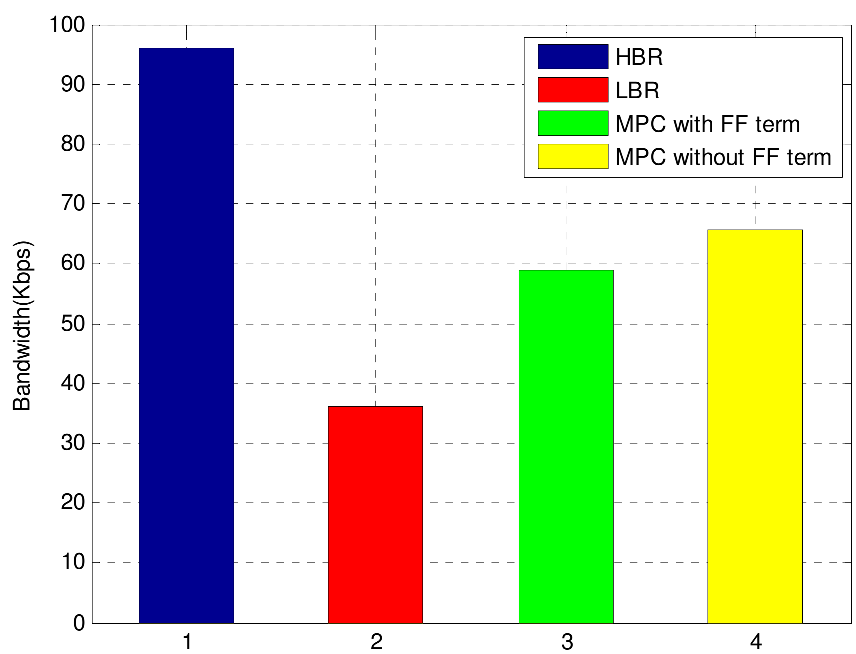

Fig. 5.12. Bandwidth used by controlled UDP flows compared to uncontrolled flows with highest (HBR) and lowest (LBR) bit rate (7% CLR).

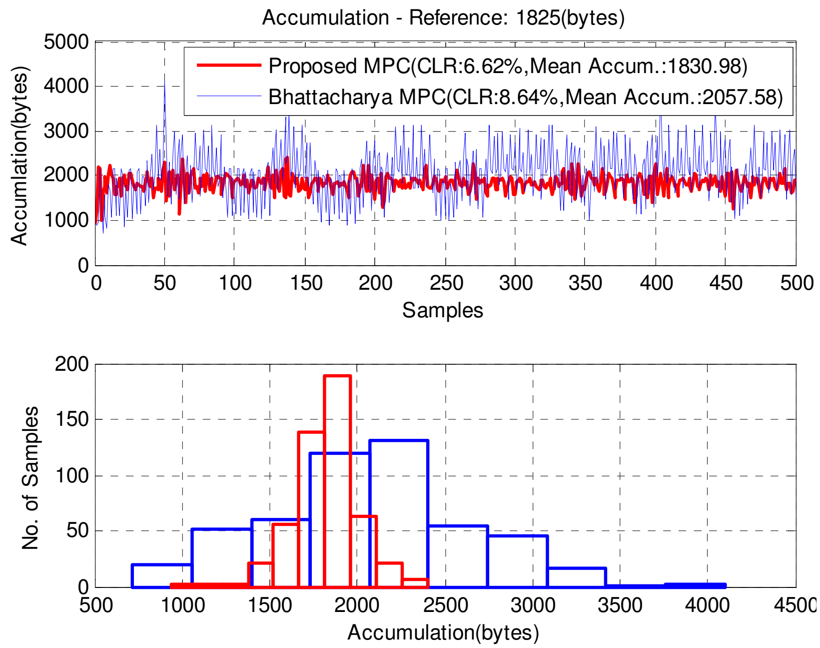

Fig. 5.13. Accumulation responses and their histogram of two MPC controllers (7% CLR).

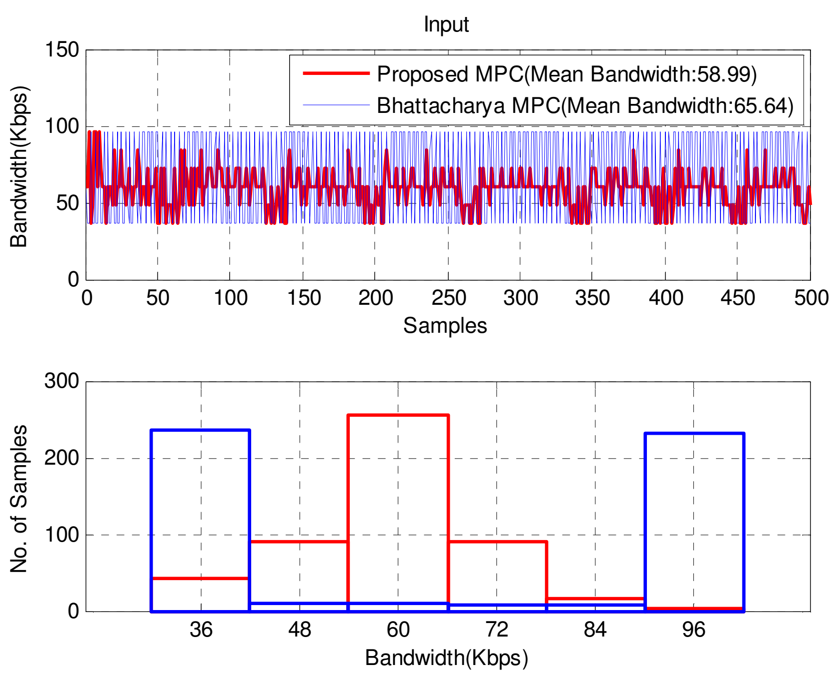

Fig. 5.14. Control inputs and their histogram of two MPC controllers (7% CLR).

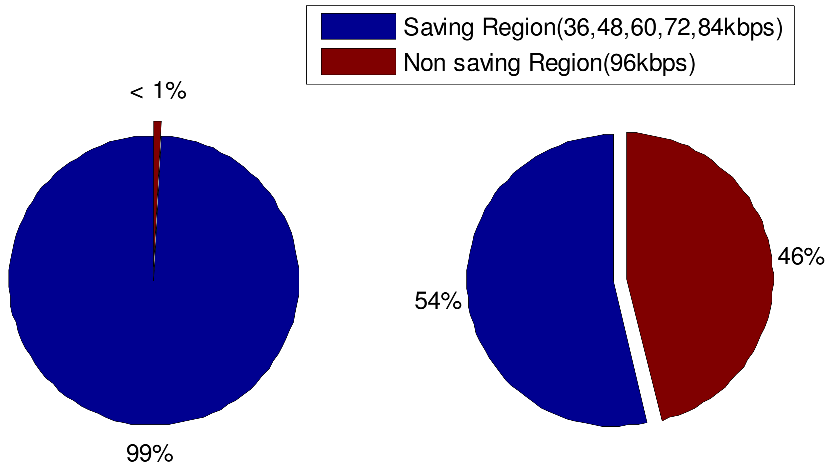

Fig. 5.15. Probability of bandwidth savings: proposed control scheme (left) and Bhattacharya scheme (right) (7% CLR).

Experimental results for the case of 9% CLR are shown in Figures 5.16-5.20. In this case, both MPC controllers still have the QoS as high as the HBR case. The proposed controller does not outperform the Bhattacharya model in term of QoS. However, it manages to save more bandwidth and has much higher probability of bandwidth savings.

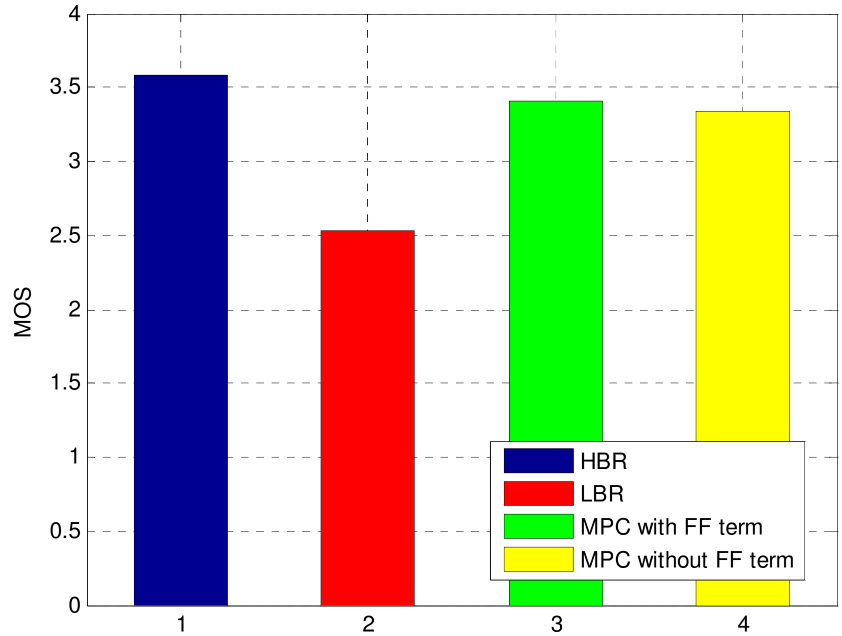

Fig. 5.16. Performance of MPC controllers in term of mean MOS of five controlled UDP flows compared to those of uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in network topology 1 with 9% CLR.

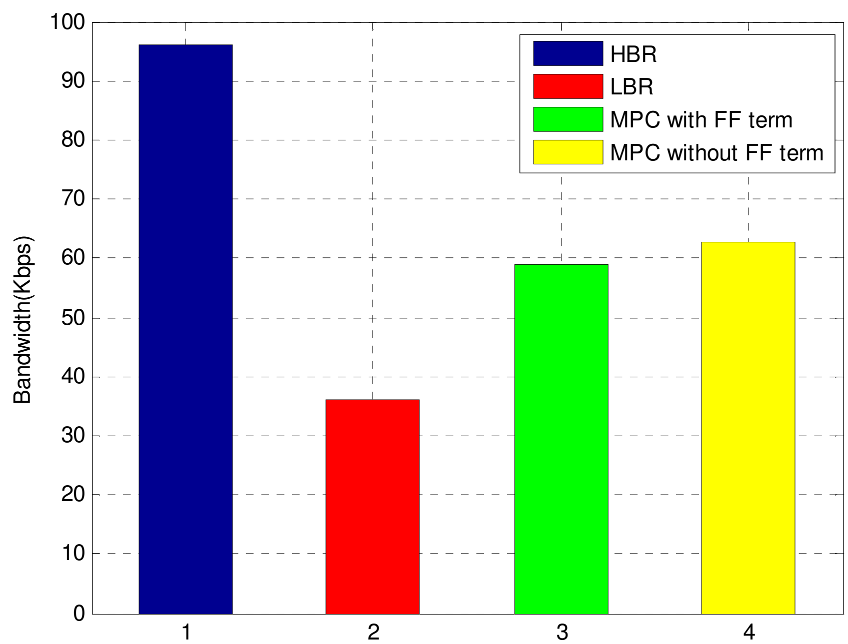

Fig. 5.17. Bandwidth used by controlled UDP flows compared to uncontrolled flows with highest (HBR) and lowest (LBR) bit rate (9% CLR).

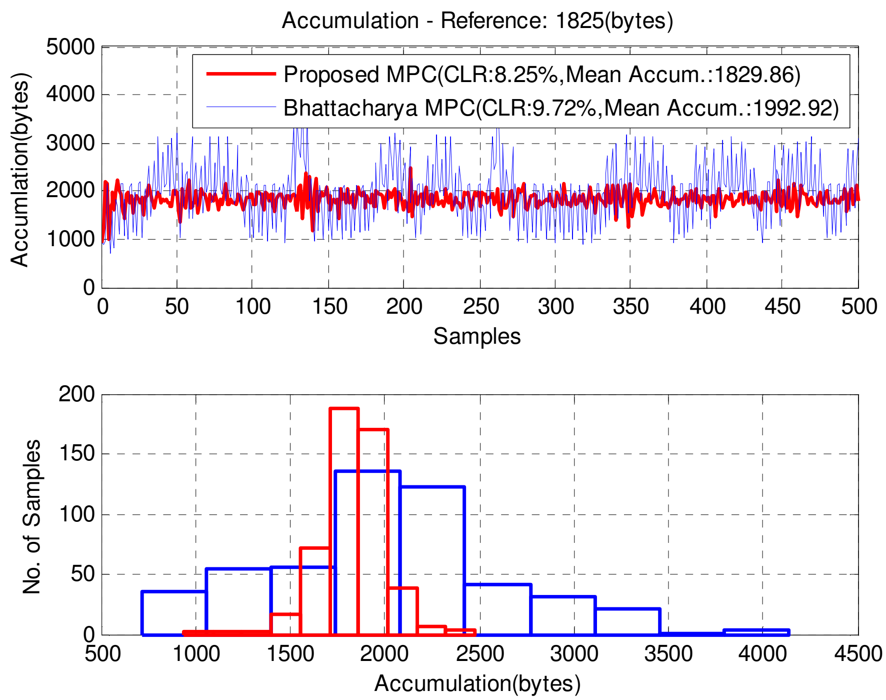

Fig. 5.18. Accumulation responses and their histogram of two MPC controllers (9% CLR).

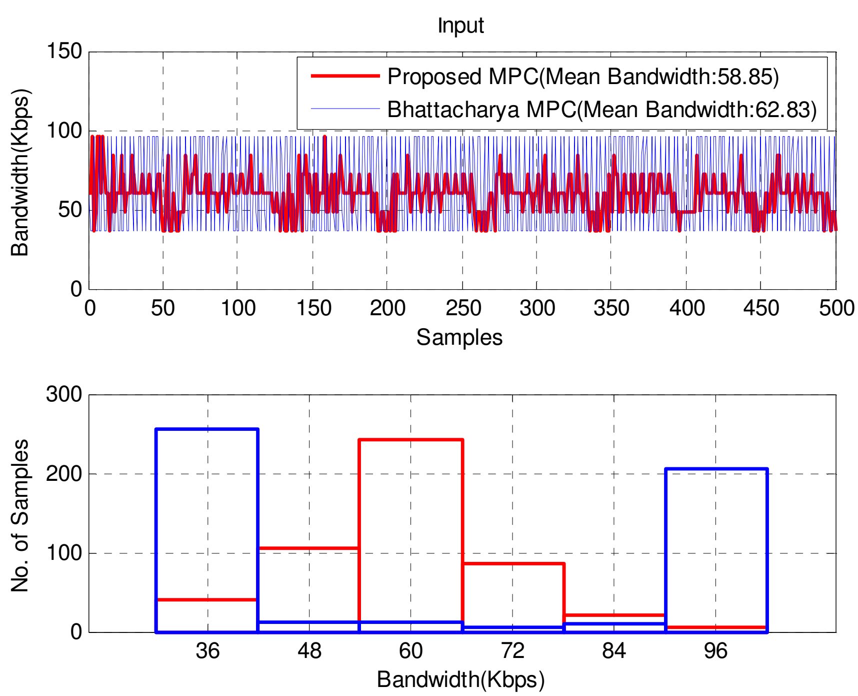

Fig. 5.19. Control inputs and their histogram of two MPC controllers (9% CLR).

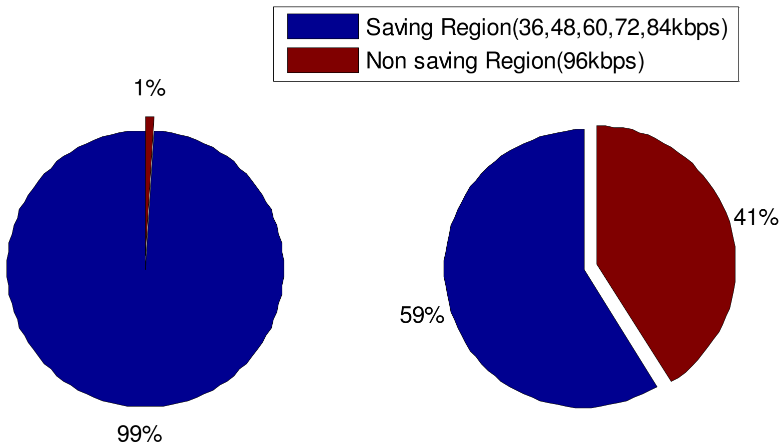

Fig. 5.20. Probability of bandwidth savings: proposed control scheme (left) and Bhattacharya scheme (right) (9% CLR).

In Figures 5.21-5.25, the CLR is increased to 11%. Again, the modified MPC controller and one proposed by Bhattacharya acquire a good QoS, compared to the case of HBR. The proposed controller with FF term saves a considerably higher bandwidth than Bhattacharya controller. In addition, it has a more stable response with a nearly certainty of bandwidth saving with 99% of P_{saving}.

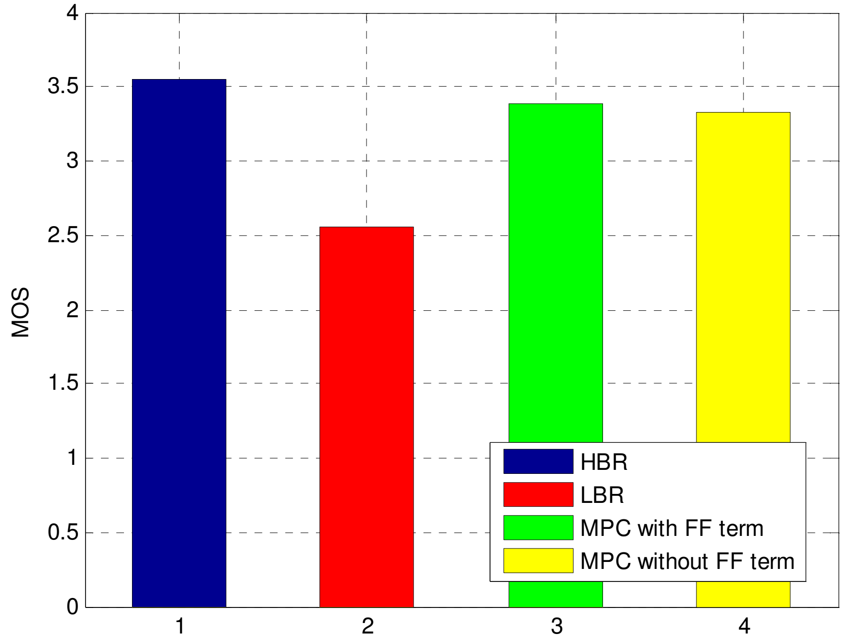

Fig. 5.21. Performance of MPC controllers in term of mean MOS of five controlled UDP flows compared to those of uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in network topology 1 with 11% CLR.

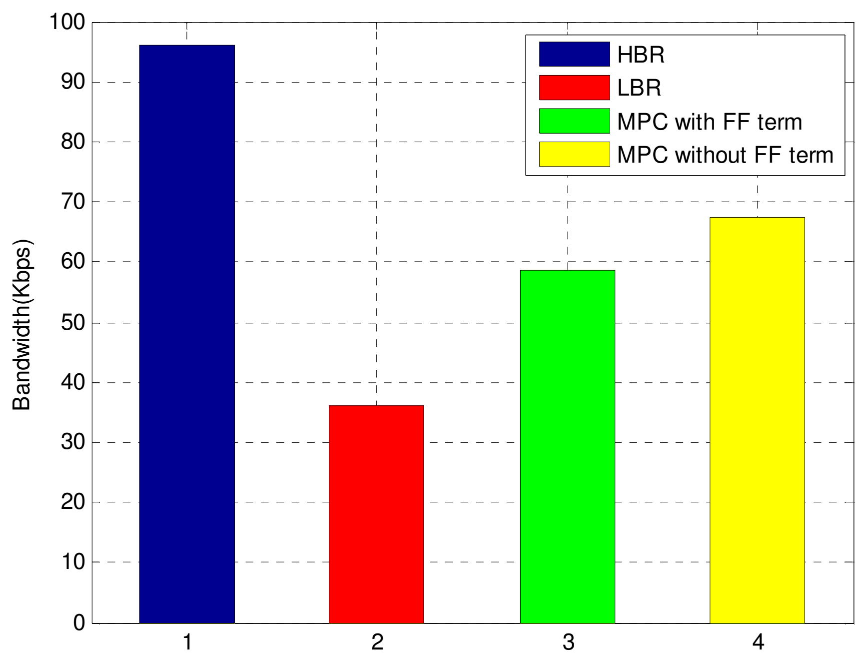

Fig. 5.22. Bandwidth used by controlled UDP flows compared to uncontrolled flows with highest (HBR) and lowest (LBR) bit rate (11% CLR).

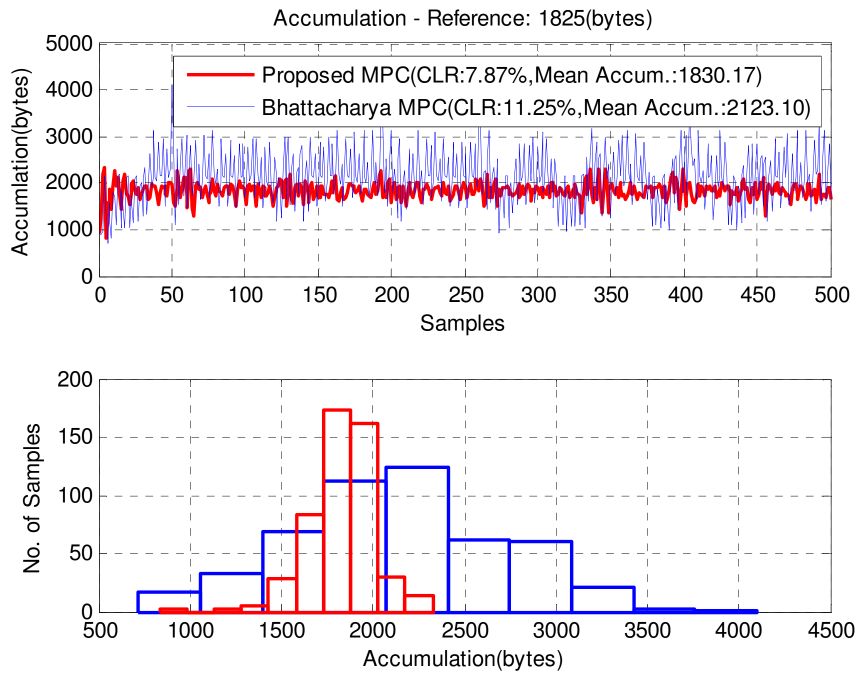

Fig. 5.23. Accumulation responses and their histogram of two MPC controllers (11% CLR).

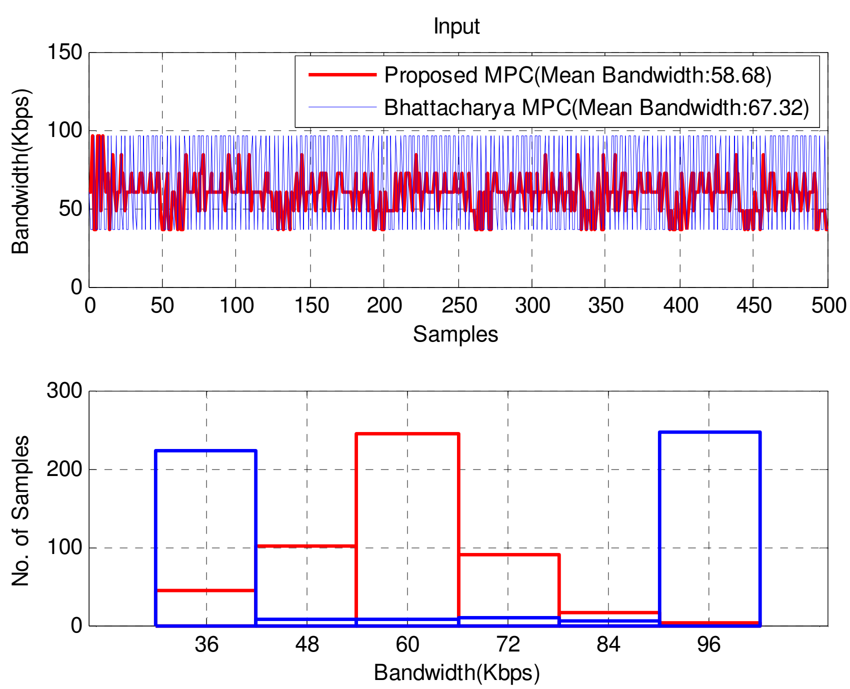

Fig. 5.24. Control inputs and their histogram of two MPC controllers (11% CLR).

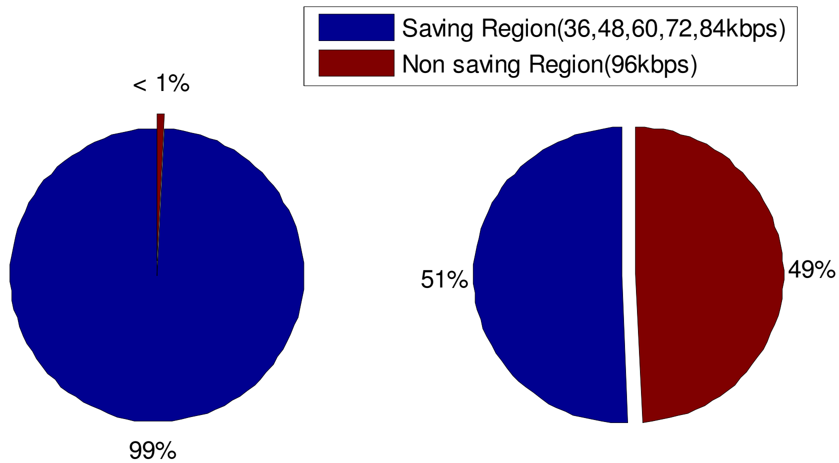

Fig. 5.25. Probability of bandwidth savings: proposed control scheme (left) and Bhattacharya scheme (right) (11% CLR).

Figures 5.26-5.30 are performances for the case of 15% CLR. It can be noticed that QoS, expressed in term of MOS, reduces as the CLR goes up high. For the loss rate of 15%, the MOS should be expected to be smaller than 3.5 unit. However, the same trend as previous cases is remained. In particular, both controllers has the QoS as high as one of the HBR case; The proposed controller has much higher probability of bandwidth savings; And, it can save more bandwidth compared to the MPC controller proposed by Bhattacharya.

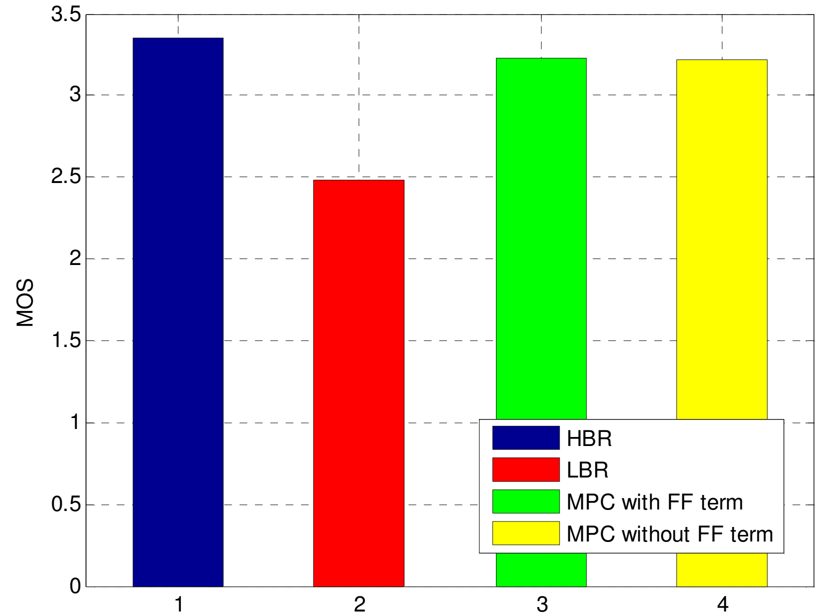

Fig. 5.26. Performance of MPC controllers in term of mean MOS of five controlled UDP flows compared to those of uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in network topology 1 with 15% CLR.

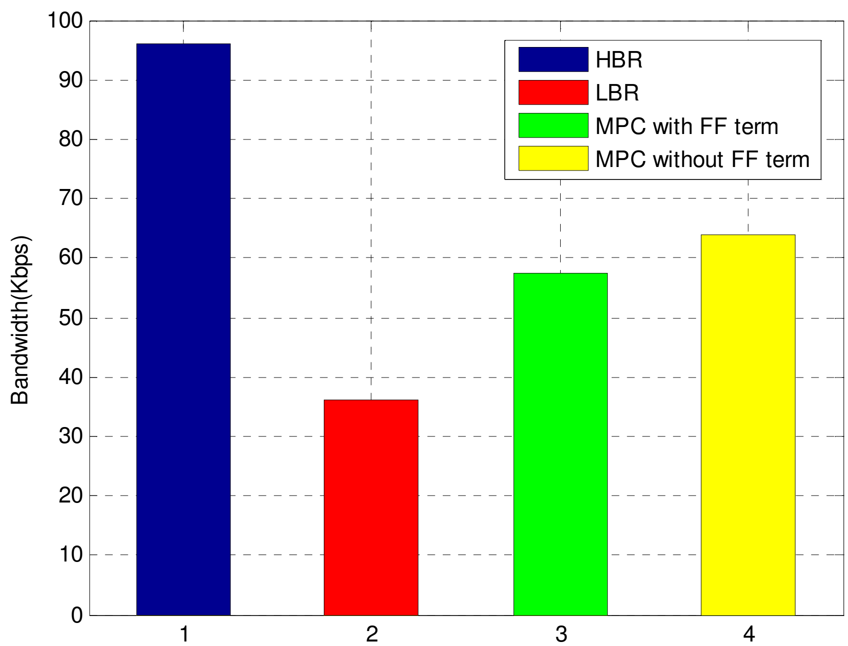

Fig. 5.27. Bandwidth used by controlled UDP flows compared to uncontrolled flows with highest (HBR) and lowest (LBR) bit rate (15% CLR).

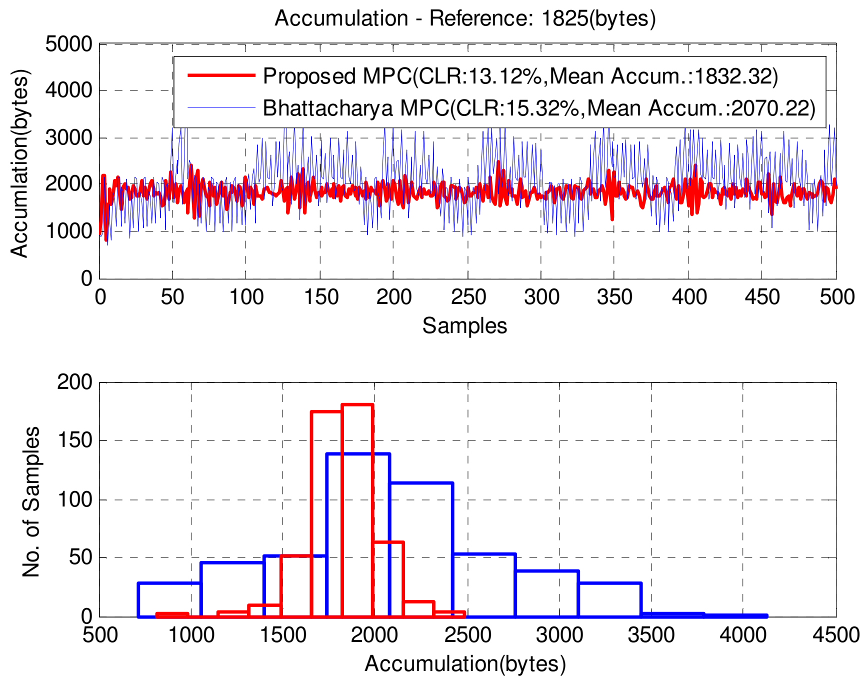

Fig. 5.28. Accumulation responses and their histogram of two MPC controllers (15% CLR).

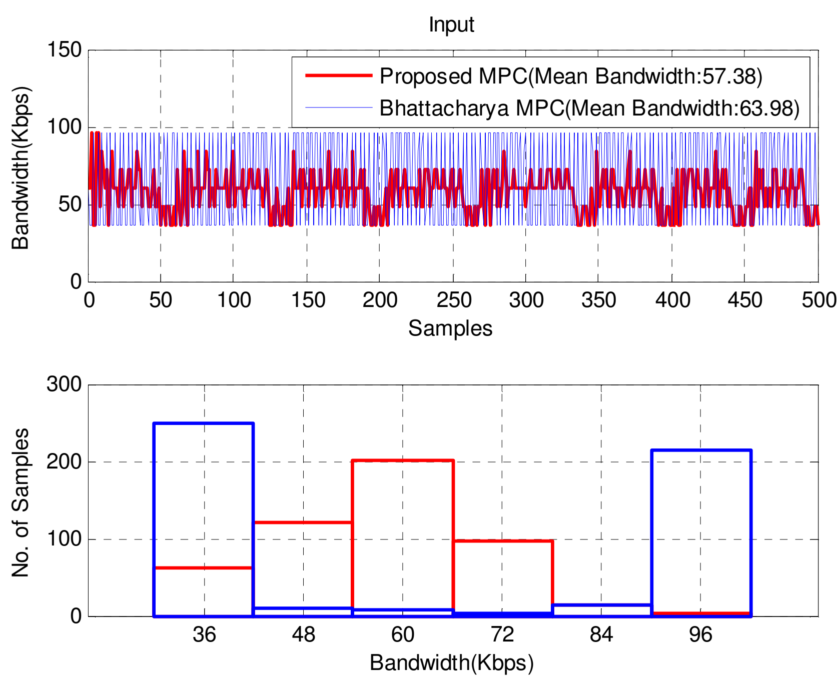

Fig. 5.29. Control inputs and their histogram of two MPC controllers (15% CLR).

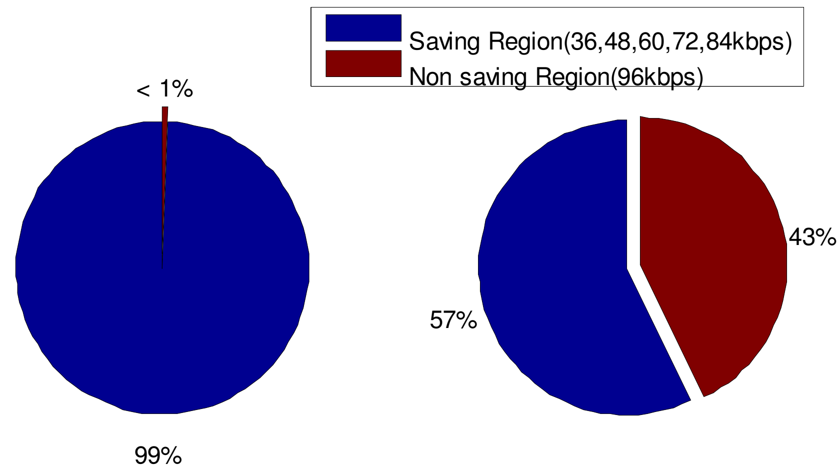

Fig. 5.30. Probability of bandwidth savings: proposed control scheme (left) and Bhattacharya scheme (right) (15% CLR).

B. Scalability of the Control Strategies

The purpose of these experiments is to see what happens when more numbers of multimedia flows in the network employ the control schemes. Two different circumstances of this increasing are considered:

(1) The share of controlled UDP flows to the total UDP traffic increases while the ratio of UDP flows and TCP flows to the total traffic remains the same:

To study the scalability of the designed control strategies in the first circumstance, four sets of experiments are conducted on network topology #2. In each set, the average bit rates of exponential (UDP) nodes are alternatively reduced. The gaining bandwidth is allocated to additional UDP nodes which apply the MPC control schemes. The number of controlled UDP applications at routers R0 and R2 increases from 5 to 36, to 72 and to 144 for each experimental case. Table XVI in Chapter II shows the variation of different network parameters corresponding to these sets of experiments.

Figures 5.31-5.35 are experimental results when the number of controlled UDP is 5. Here, there exists the same trend as in the previous experiments of which loss rates were varied. In specific, both MPC controllers gain good QoS and help to save bandwidth usage; the proposed controller has a more stable response and a much higher probability of bandwidth savings compared to Bhattacharya controller.

One may notice that this experiment is similar to the previous one of 3% CLR case. In fact, both of the experiments have the same 5 controlled UDP and 3% CLR; the CLR is keep constant around 3% for all experiments conducted on topology #2. However, the MOS for this case is smaller than that of the previous case. This is due to the difference between two network topologies. The fact of similar performances between two experiments with a slight reduction for the current one much or less shows the robustness of the MPC controllers.

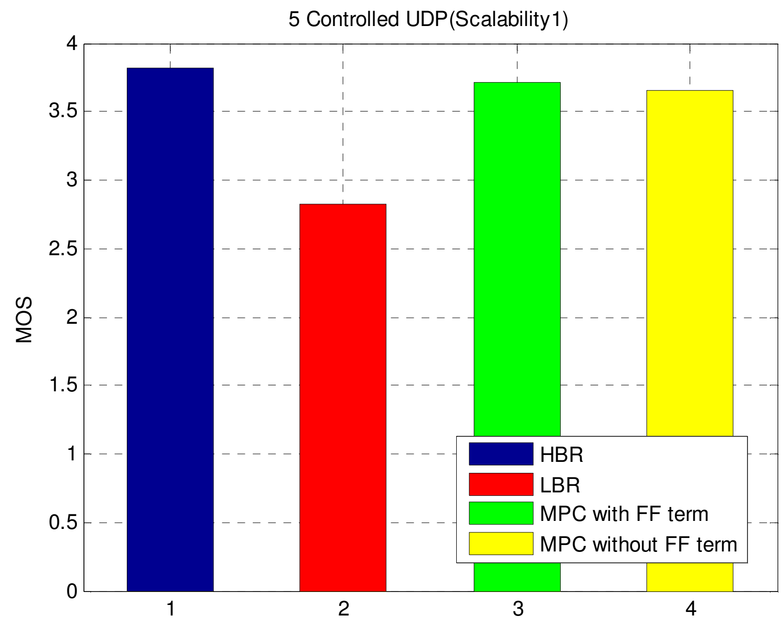

Fig. 5.31. Scalability performance of MPC controllers in term of mean MOS of 5 controlled UDP flows compared to those of uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in the first circumstance.

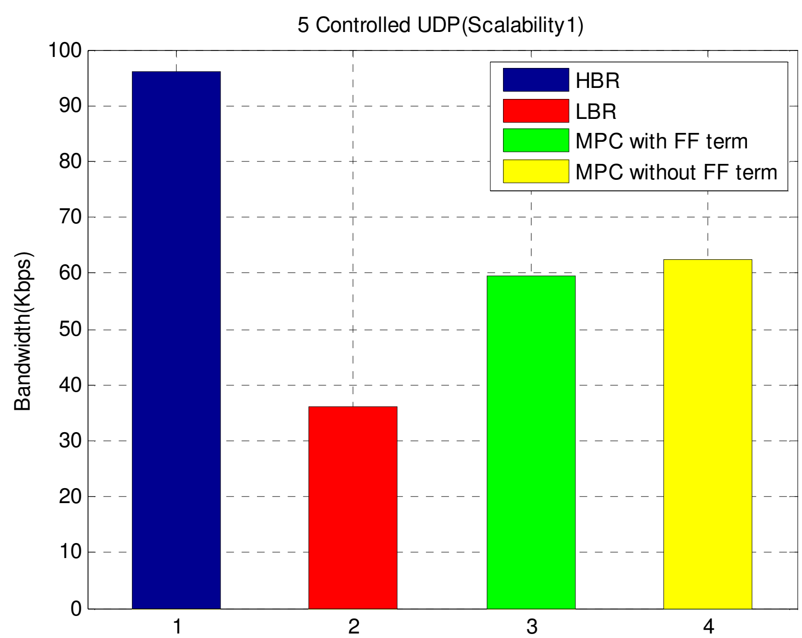

Fig. 5.32. Bandwidth used by controlled UDP flows compared to uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in the first scalability aspect (5 controlled UDP).

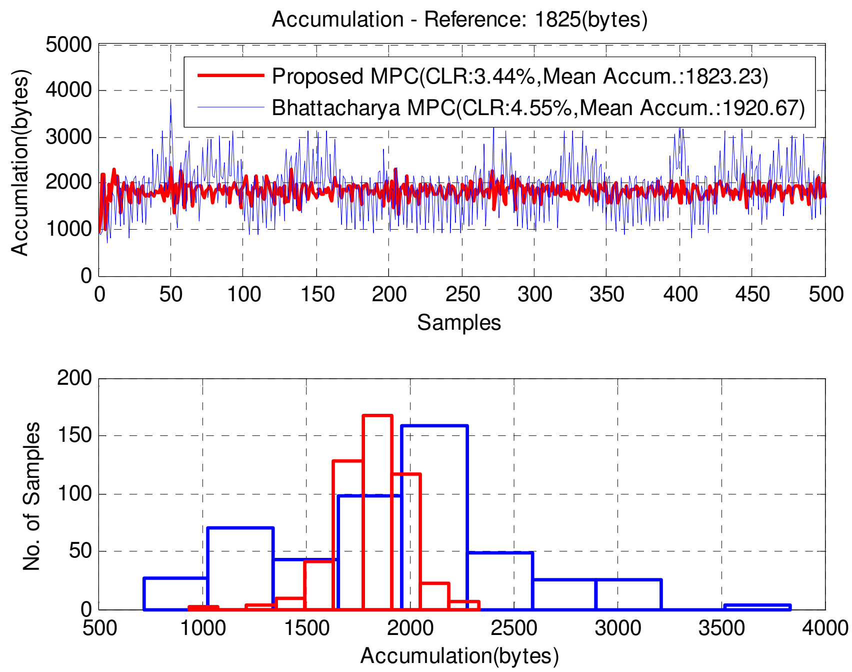

Fig. 5.33. Accumulation responses and their histogram of two MPC controllers (5 controlled UDP).

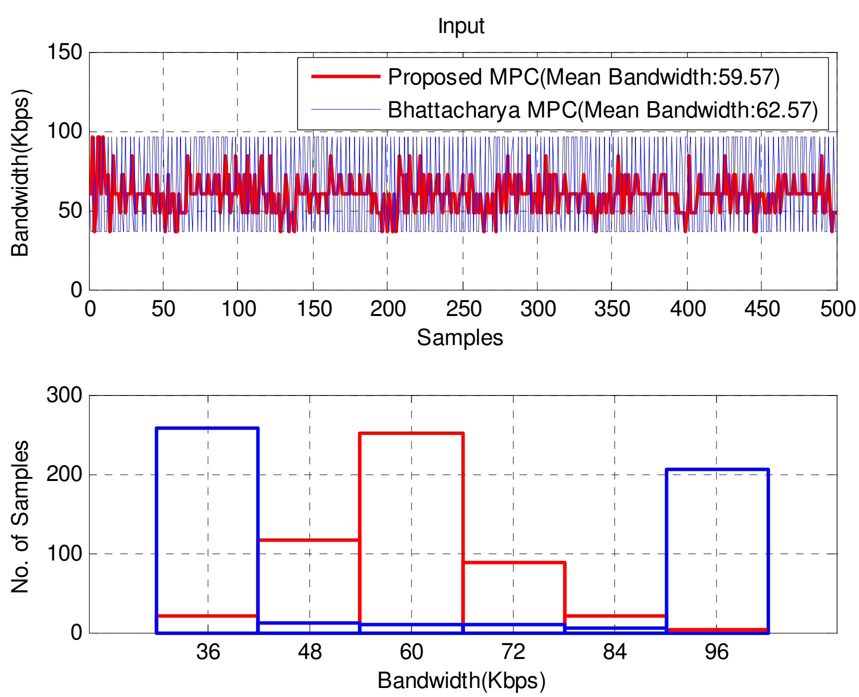

Fig. 5.34. Control inputs and their histogram of two MPC controllers (5 controlled UDP).

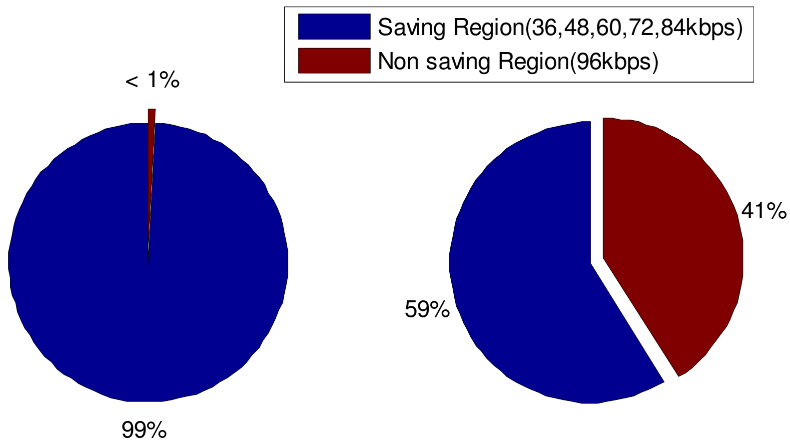

Fig. 5.35. Probability of bandwidth savings: proposed control scheme (left) and Bhattacharya scheme (right) (5 controlled UDP).

Figures 5.36-5.40 are experimental results when the number of controlled UDP is increased to 36. With the CLR fixed around 3%, the QoS does not change much as the number of controlled UDP goes up. It can be seen from the figures that both of the controllers can scale up well to the increase. In particular, the features of good QoS, bandwidth saving and probability of bandwidth savings are remained in the same trend.

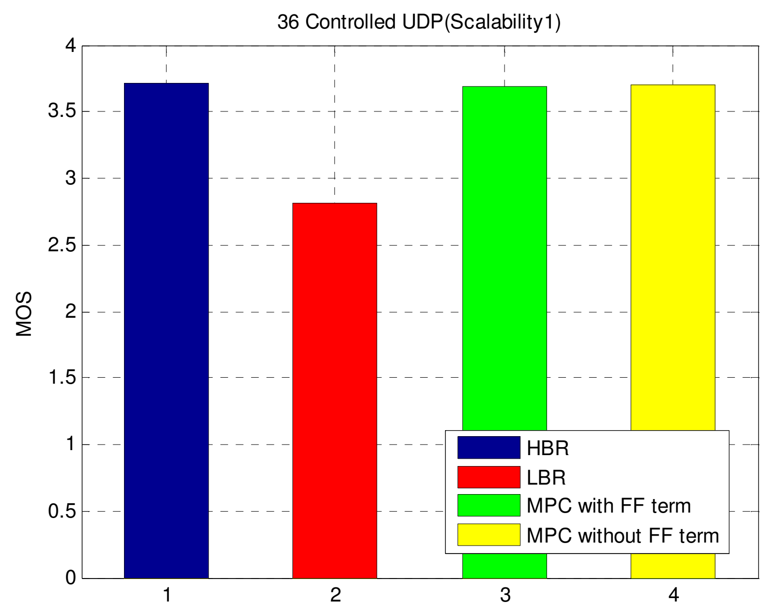

Fig. 5.36. Scalability performance of MPC controllers in term of mean MOS of 36 controlled UDP flows compared to those of uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in the first circumstance.

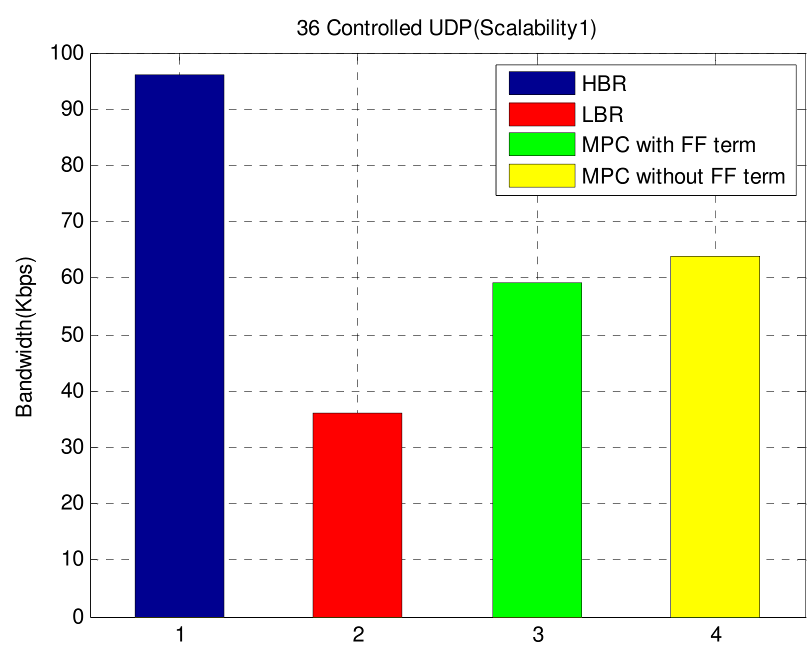

Fig. 5.37. Bandwidth used by controlled UDP flows compared to uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in the first scalability aspect (36 controlled UDP).

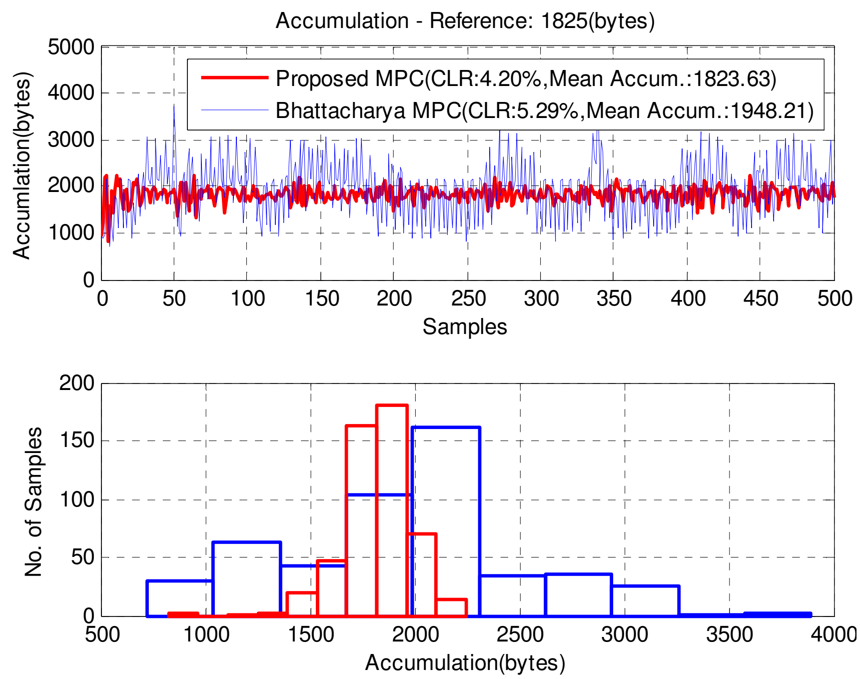

Fig. 5.38. Accumulation responses and their histogram of two MPC controllers (36 controlled UDP).

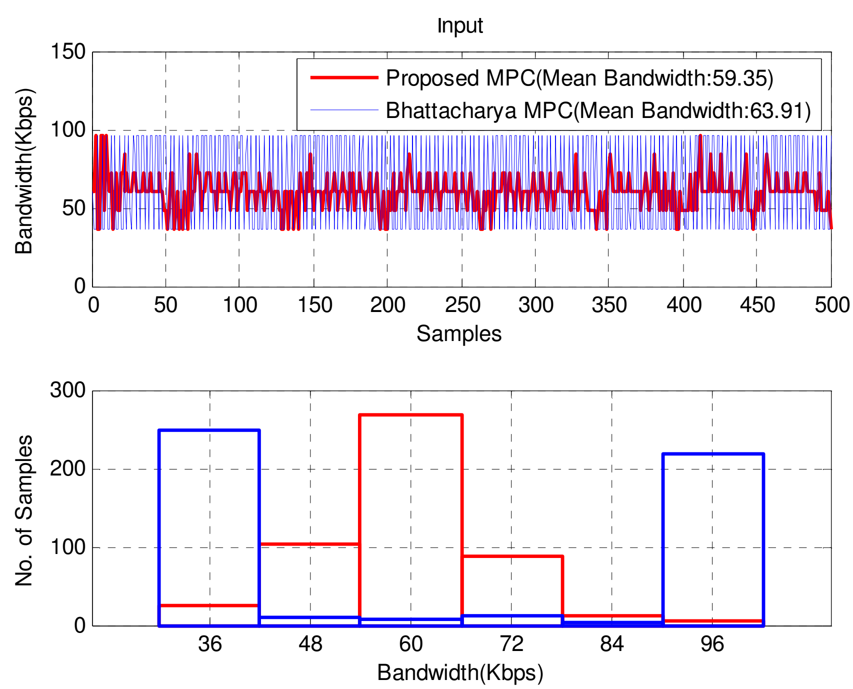

Fig. 5.39. Control inputs and their histogram of two MPC controllers (36 controlled UDP).

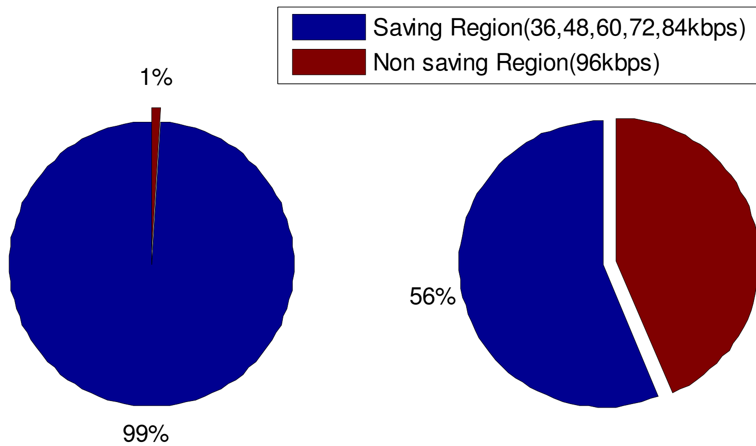

Fig. 5.40. Probability of bandwidth savings: proposed control scheme (left) and Bhattacharya scheme (right) (36 controlled UDP).

As the number of controlled UDP increases to 72, the MPC controllers are still able to scale up well, as shown in Figures 5.41-5.45. Again, the MPC controllers acquires the QoS as good as one of the HBR case. The probability of bandwidth savings of the modified controller is about 99% compared to 58% for the MPC proposed by Bhattacharya.

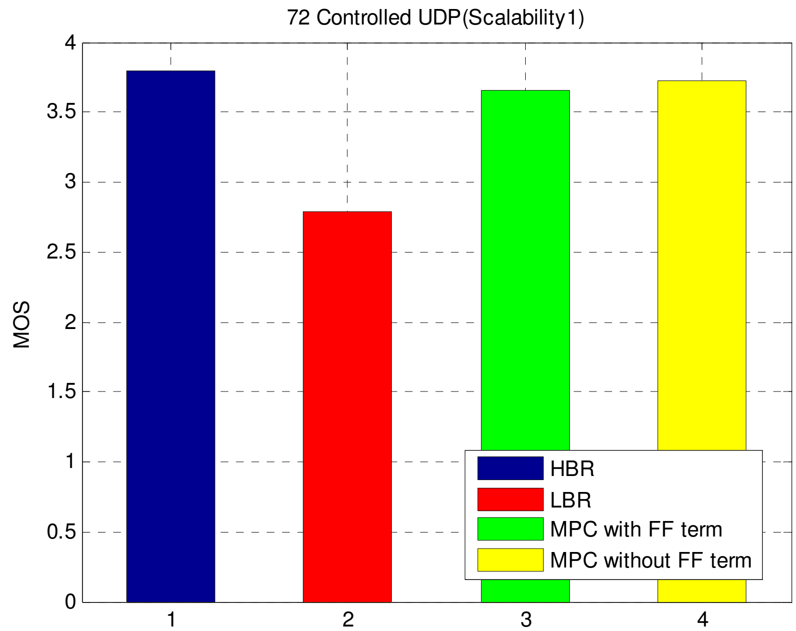

Fig. 5.41. Scalability performance of MPC controllers in term of mean MOS of 72 controlled UDP flows compared to those of uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in the first circumstance.

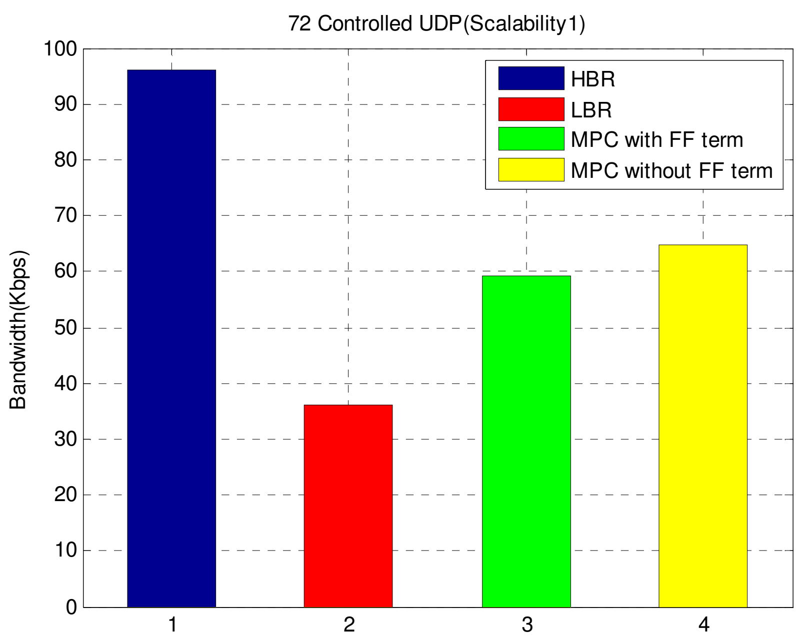

Fig. 5.42. Bandwidth used by controlled UDP flows compared to uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in the first scalability aspect (72 controlled UDP).

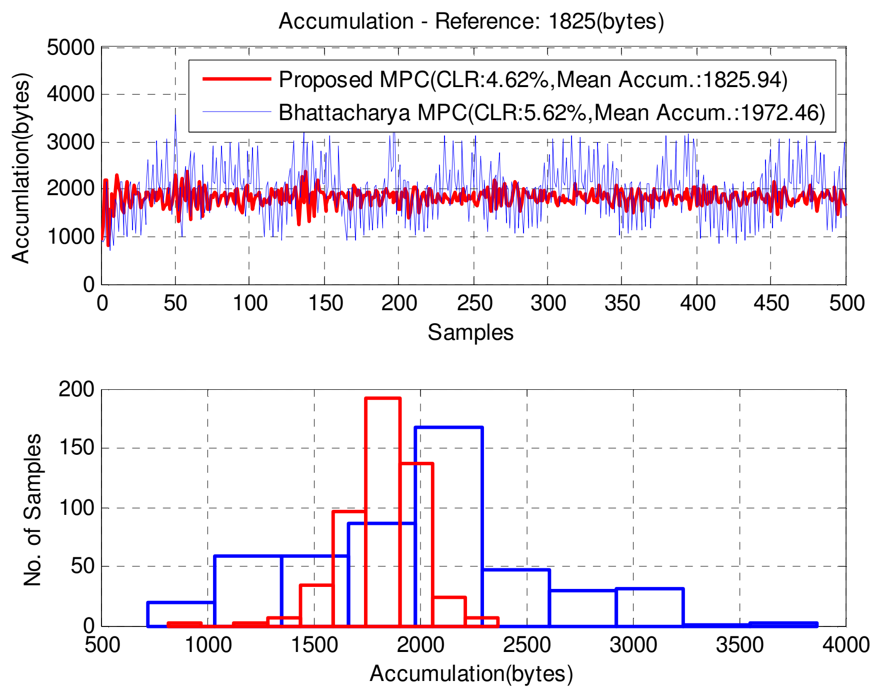

Fig. 5.43. Accumulation responses and their histogram of two MPC controllers (72 controlled UDP).

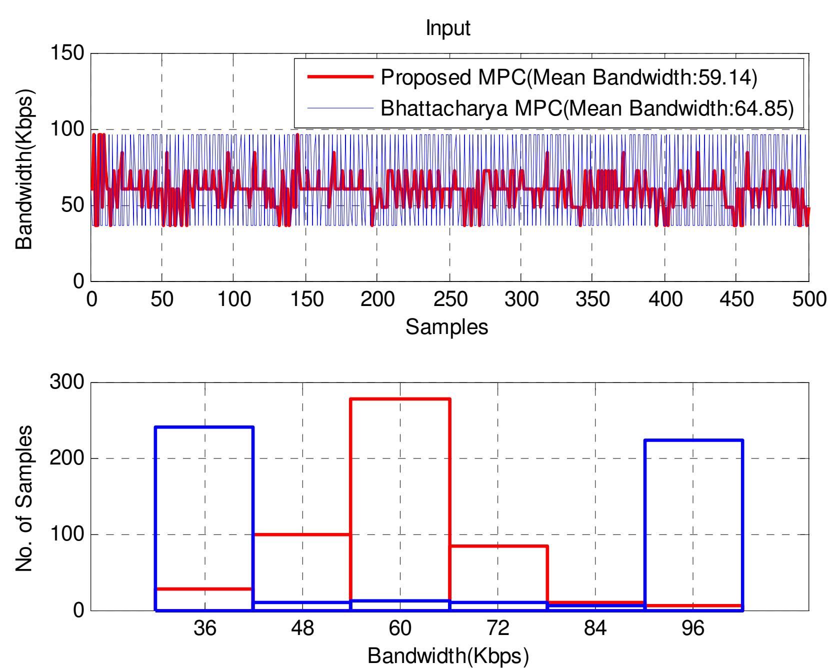

Fig. 5.44. Control inputs and their histogram of two MPC controllers (72 controlled UDP).

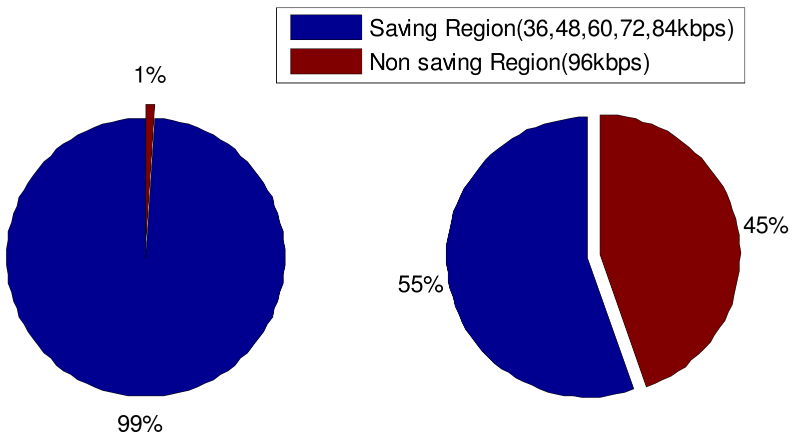

Fig. 5.45. Probability of bandwidth savings: proposed control scheme (left) and Bhattacharya scheme (right) (72 controlled UDP).

Figures 5.46-5.50 conclude the experiments for the scalability test of which the share of controlled UDP flows to the total UDP flows increases while the share of total UDP flows to the total trafic is kept constant. The similar performances between this case of 144 controlled UDP and the case of 5 controlled UDP shows that the MPC strategies, indeed, are able to scale up well to this type of scalability.

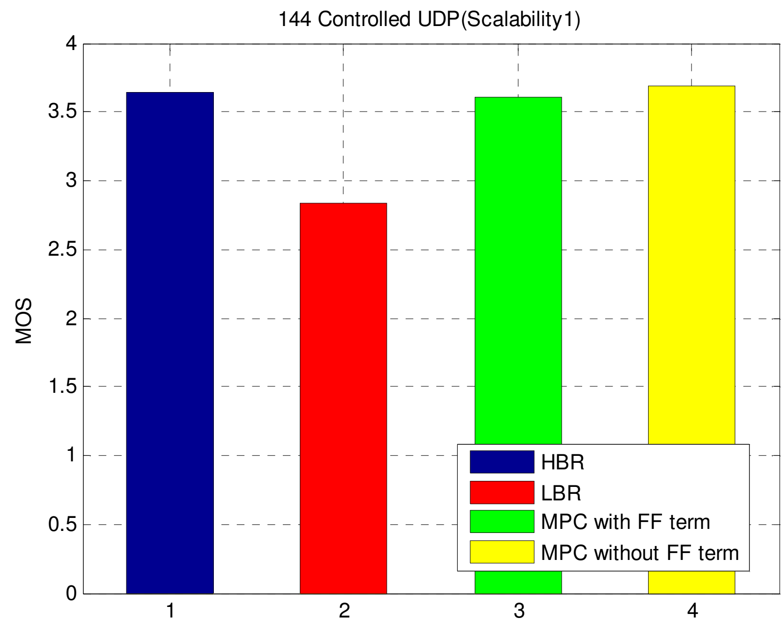

Fig. 5.46. Scalability performance of MPC controllers in term of mean MOS of 144 controlled UDP flows compared to those of uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in the first circumstance.

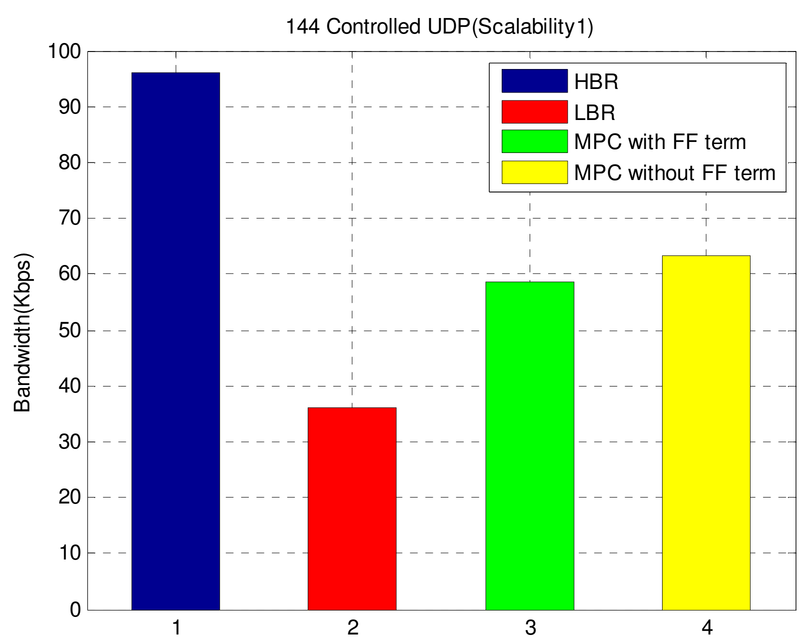

Fig. 5.47. Bandwidth used by controlled UDP flows compared to uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in the first scalability aspect (144 controlled UDP).

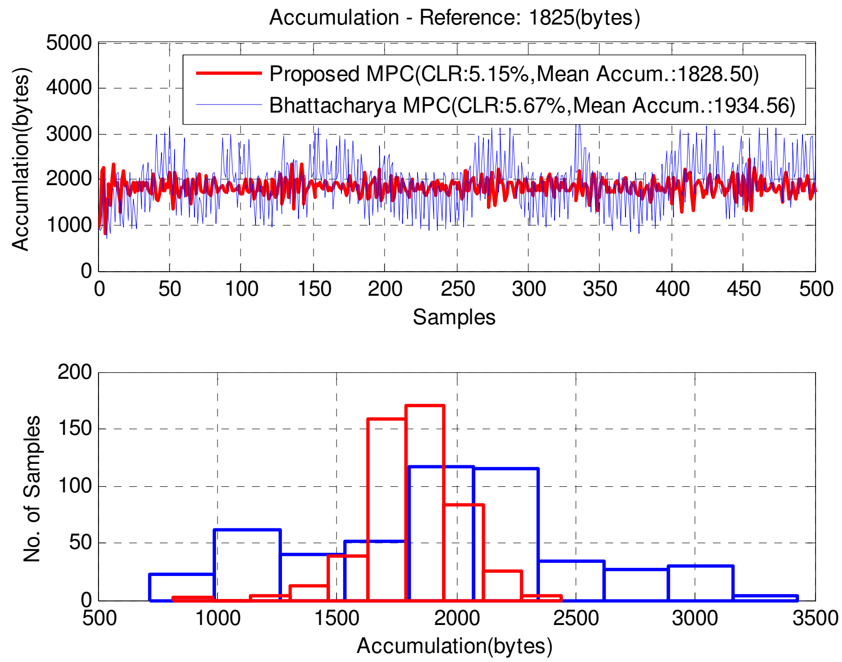

Fig. 5.48. Accumulation responses and their histogram of two MPC controllers (144 controlled UDP).

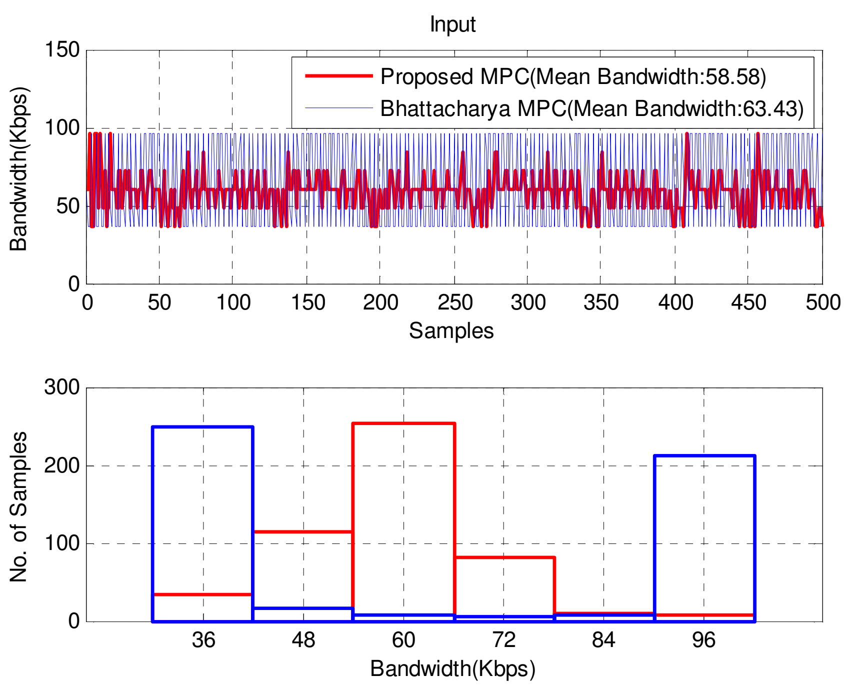

Fig. 5.49. Control inputs and their histogram of two MPC controllers (144 controlled UDP).

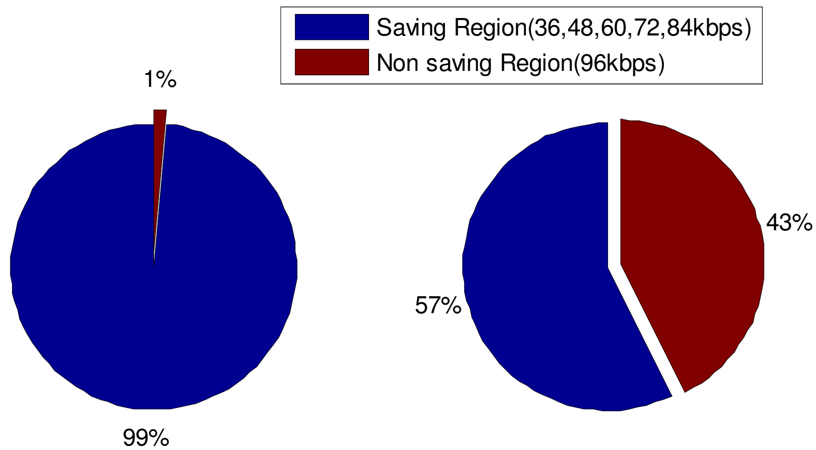

Fig. 5.50. Probability of bandwidth savings: proposed control scheme (left) and Bhattacharya scheme (right) (144 controlled UDP).

(2) The share of total UDP flows to the total traffic grows up:

In this circumstance, percentage of the controlled UDP traffic in the total traffic is increased. This is achieved by adding more controlled UDP nodes at two ends of network topology while keeping other parameters of the second topology unchanged. There are also four sets of experiments in this case. In each set, the number of controlled UDP nodes increases from 5 to 20, to 50 and to 80. Corresponding to the increase of UDP flows in percentage, the ratio of TCP flows to total traffic decrease. Here, the 5 controlled UDP case is the same as one performed in the first circumstance.

Performances of the control strategies, in case of 20 controlled UDP nodes, are shown in Figures 5.51-5.55. Again, both MPC schemes work well in the second circumstance of the scalability test. Since the CLR is kept constant around 3% in network topology #2, there is not any significant change in QoS of the scalability tests. The proposed controller still has a very high probability of bandwidth savings.

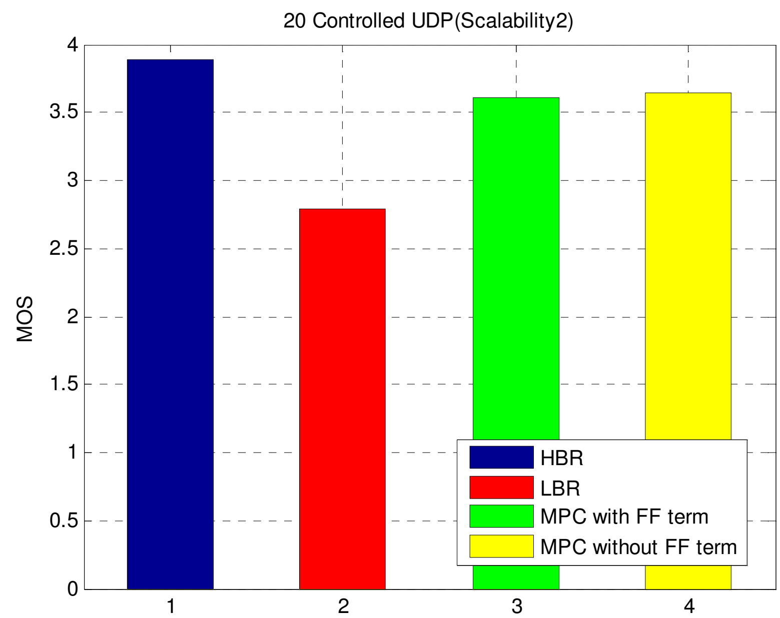

Fig. 5.51. Scalability performance of MPC controllers in term of mean MOS of 20 controlled UDP flows compared to those of uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in the second circumstance.

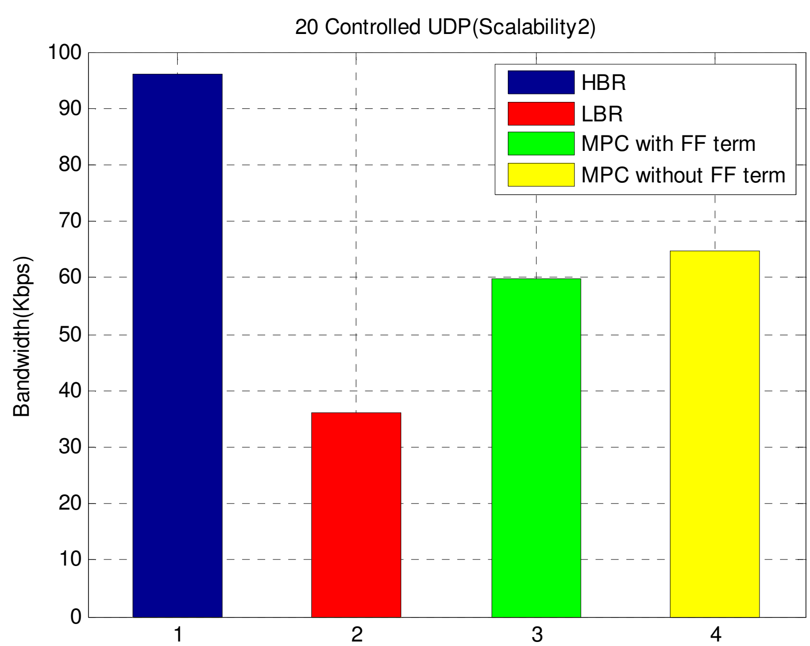

Fig. 5.52. Bandwidth used by controlled UDP flows compared to uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in the second scalability aspect (20 controlled UDP).

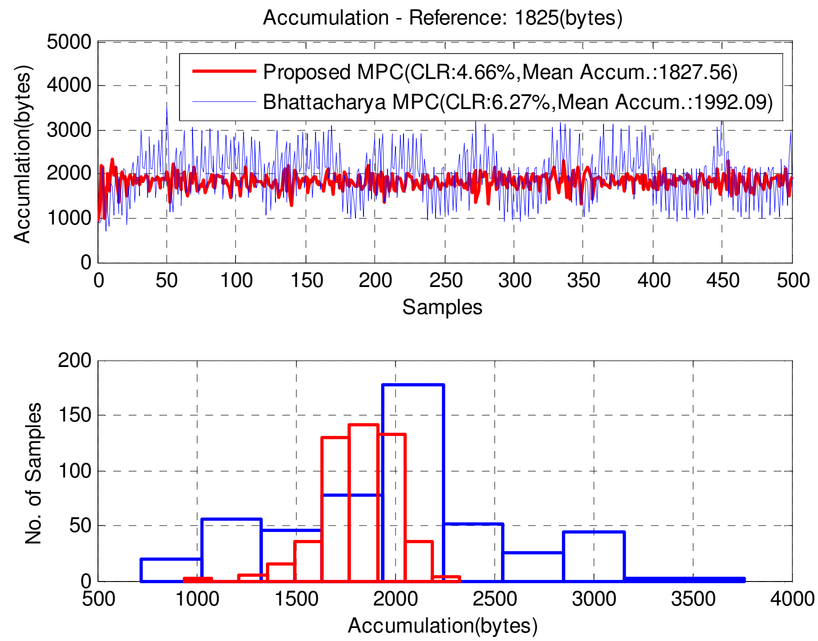

Fig. 5.53. Accumulation responses and their histogram of two MPC controllers (20 controlled UDP).

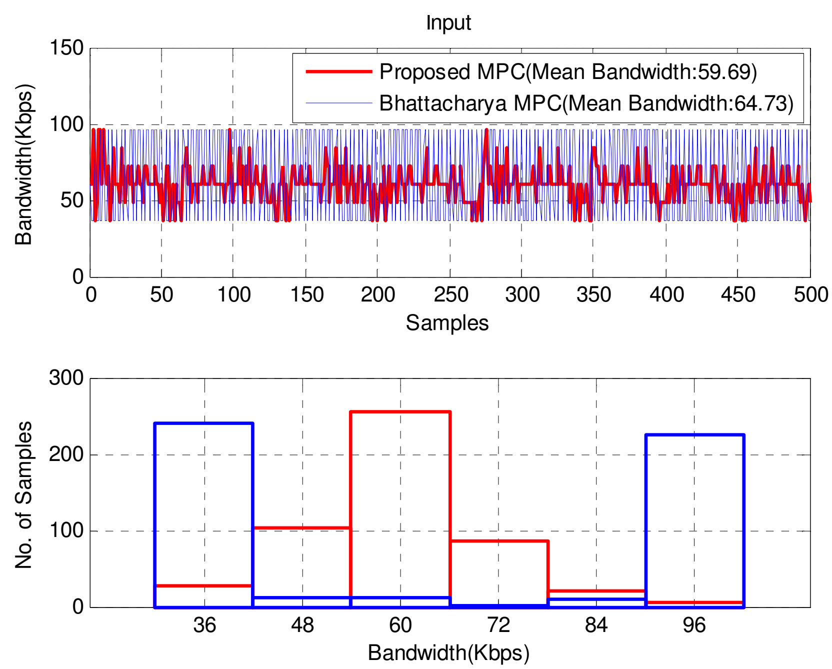

Fig. 5.54. Control inputs and their histogram of two MPC controllers (20 controlled UDP).

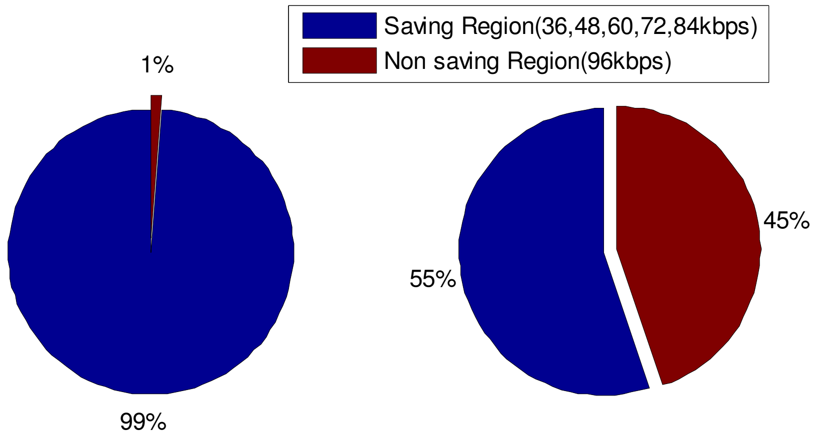

Fig. 5.55. Probability of bandwidth savings: proposed control scheme (left) and Bhattacharya scheme (right) (20 controlled UDP).

Figures 5.56-5.60 are experimental results for the case of 50 controlled UDP nodes. Similarly, the controllers show their capability of scaling well to the increase of the total UDP to the total traffic by remaining their benefits from previous experiments.

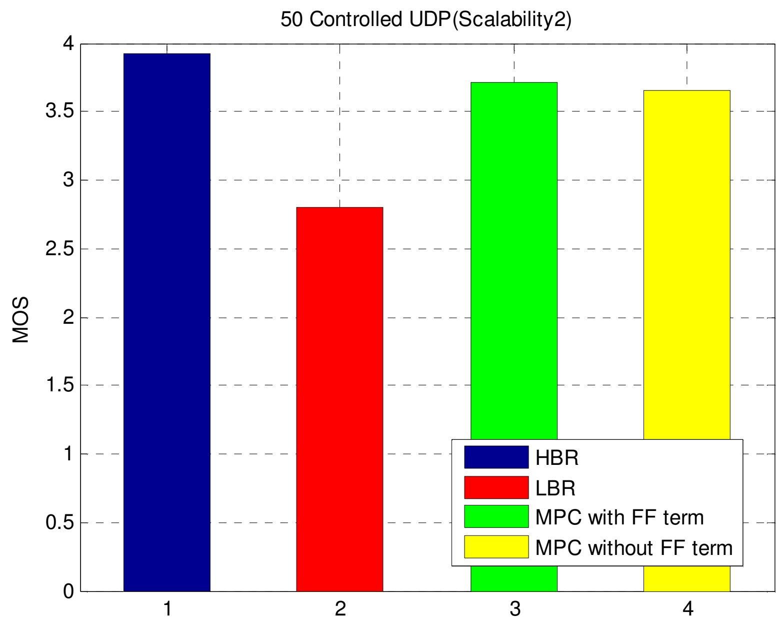

Fig. 5.56. Scalability performance of MPC controllers in term of mean MOS of 50 controlled UDP flows compared to those of uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in the second circumstance.

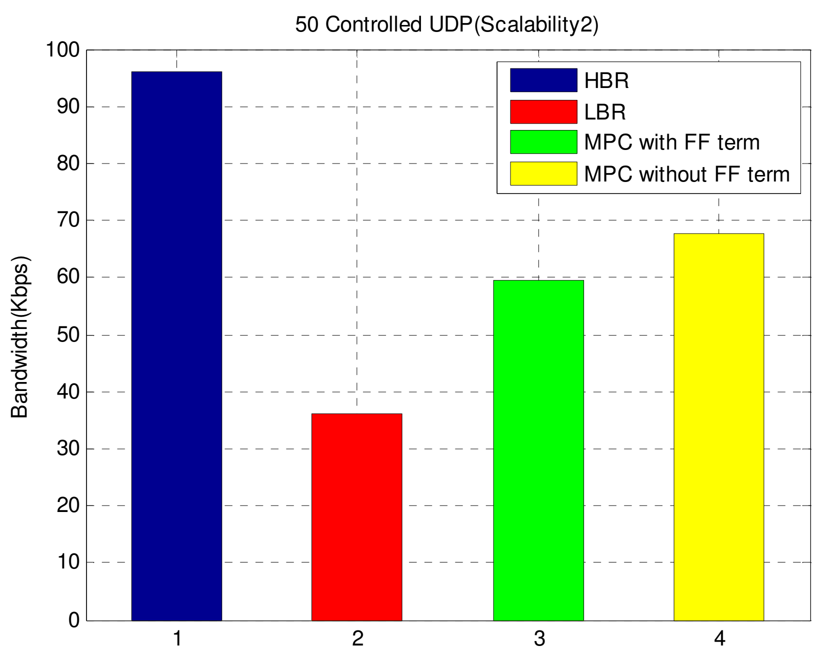

Fig. 5.57. Bandwidth used by controlled UDP flows compared to uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in the second scalability aspect (50 controlled UDP).

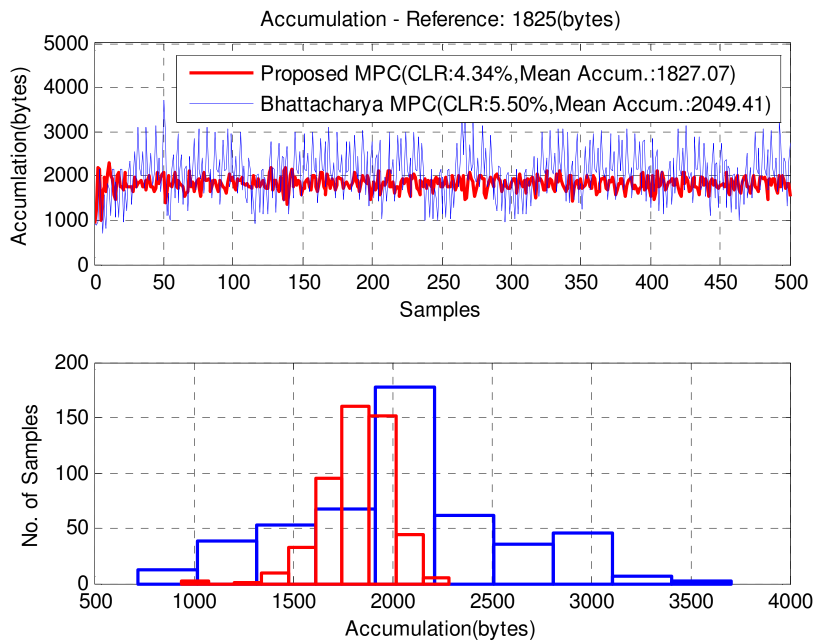

Fig. 5.58. Accumulation responses and their histogram of two MPC controllers (50 controlled UDP).

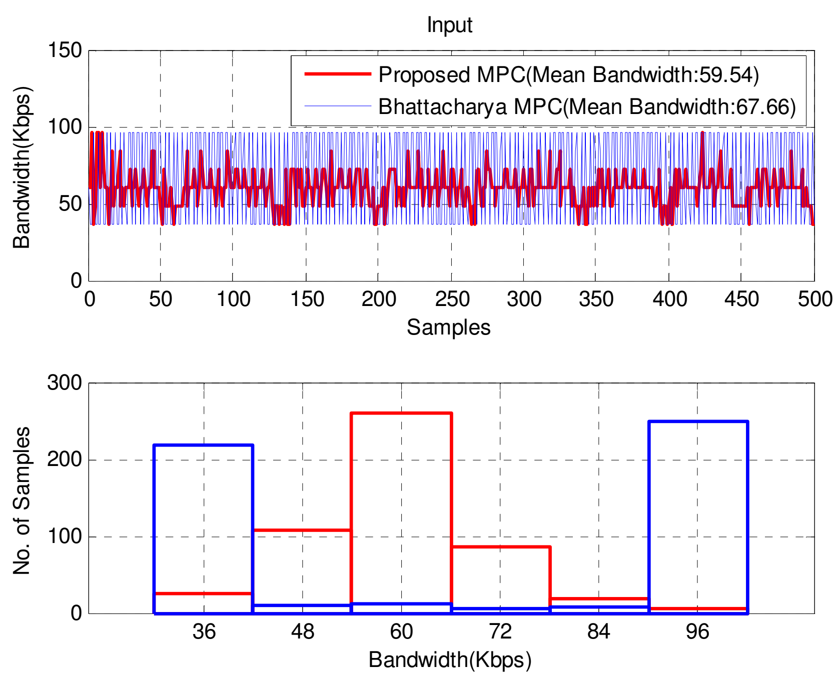

Fig. 5.59. Control inputs and their histogram of two MPC controllers (50 controlled UDP).

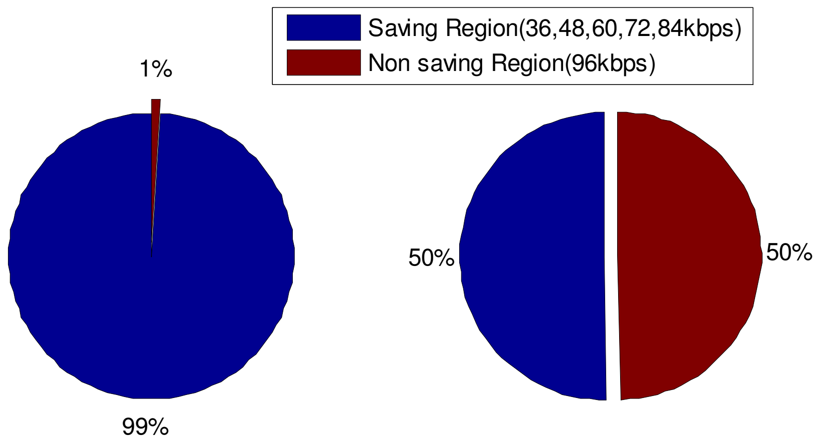

Fig. 5.60. Probability of bandwidth savings: proposed control scheme (left) and Bhattacharya scheme (right) (50 controlled UDP).

Experimental results for the case of 80 controlled UDP nodes, Figures 5.60-5.65, verified that the MPC strategies can scale up well to the increase in the ratio of UDP flows to the total traffic.

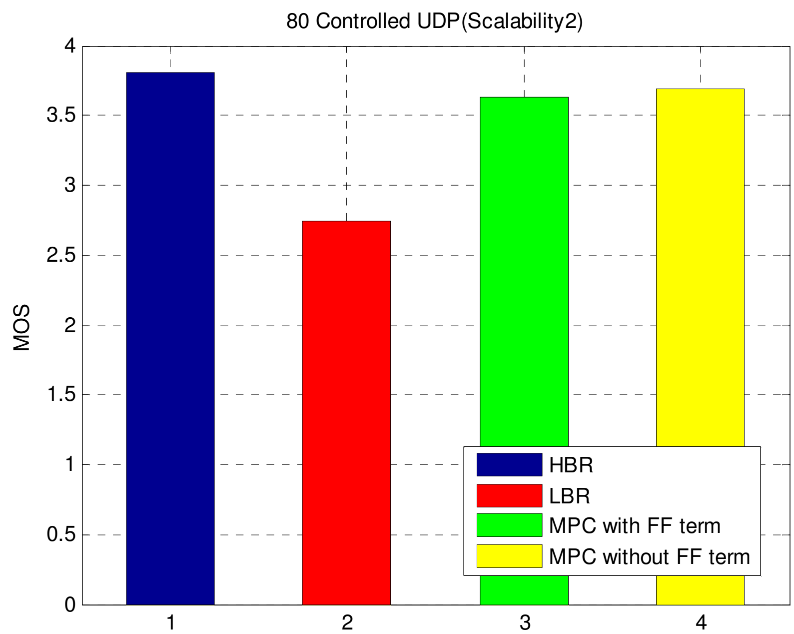

Fig. 5.61. Scalability performance of MPC controllers in term of mean MOS of 80 controlled UDP flows compared to those of uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in the second circumstance.

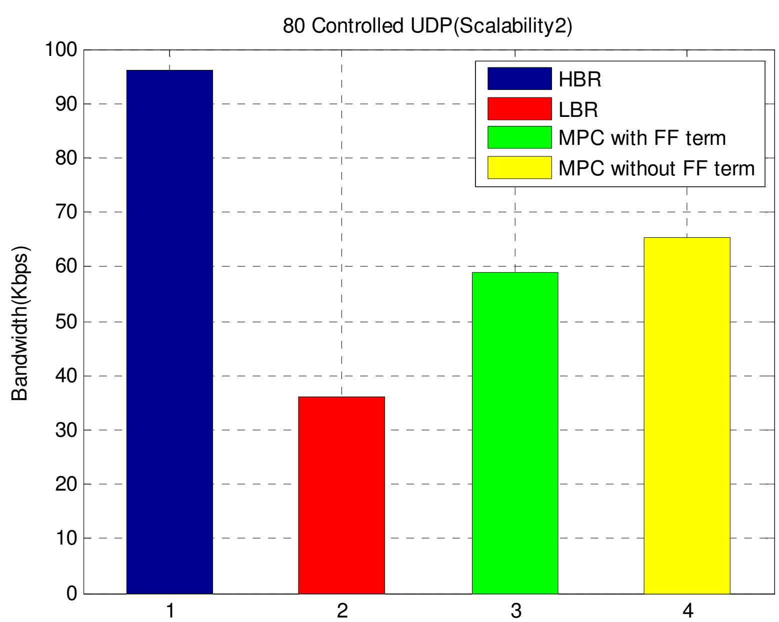

Fig. 5.62. Bandwidth used by controlled UDP flows compared to uncontrolled flows with highest (HBR) and lowest (LBR) bit rate in the second scalability aspect (80 controlled UDP).

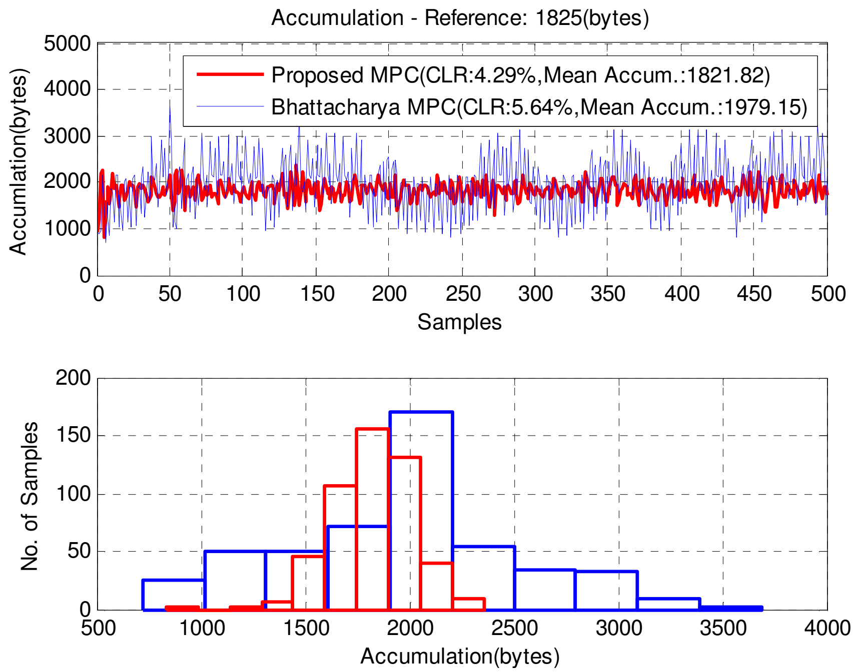

Fig. 5.63. Accumulation responses and their histogram of two MPC controllers (80 controlled UDP).

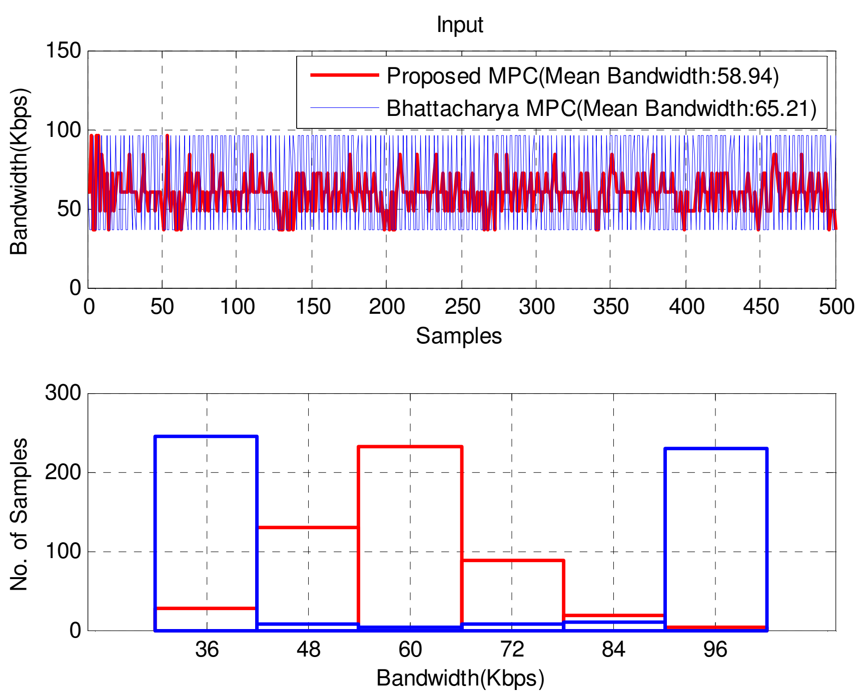

Fig. 5.64. Control inputs and their histogram of two MPC controllers (80 controlled UDP).

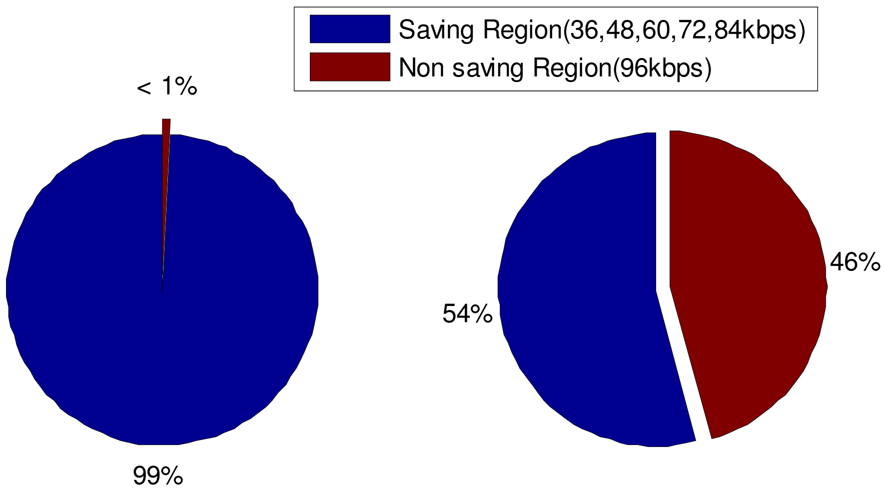

Fig. 5.65. Probability of bandwidth savings: proposed control scheme (left) and Bhattacharya scheme (right) (80 controlled UDP).

CHAPTER VI

SUMMARY AND CONCLUSION

A. Summary

Many researches have been conducted to guarantee the QoS of online multimedia services. Among them, the model predictive controller proposed by Aninda Bhattacharya is a promising approach. The strategy is able to gain good QoS in addition with the capability to save bandwidth usage. However, the deficiency, of which the bandwidth usage jumps frequently between highest and lowest values, exists within the work. To eliminate this deficiency, the current research proposes to add a feed forward term in addition to feedback terms to Bhattacharya MPC strategy. There were two network topologies constructed: (1) one for designing and validating the model predictive controllers and (2) one for studying the scalability of the designed controllers. Various simulation scenarios were carried on on two network topologies to see: (1) the effect of MPC controllers under different loss rates and (2) the scalability of the controllers when the number of controlled UDP nodes increases.

B. Conclusion

Experimental results show that both controllers manage to gain good QoS while saving bandwidth usage of the multimedia flows. Also, they are able to scale up well to different aspects of the scalability tests. Compared to Bhattacharya model, the proposal MPC strategy shows the advantages of being more stable in response, saving more bandwidth usage and possessing a much higher probability of bandwidth savings. In particular, its probability of bandwidth savings is up to about 99% while Bhattacharya model gives only about 58%. The practical meaning of a controller possessing 100% probability of bandwidth saving is that it can always save resource for the network.

C. Limitations

The proposed MPC strategy helps to improve the control results. However, it can not solve existing limitations cited out in Bhattacharya's research:

- The lack of real world test bed: even though experimental results are impressive, their correctness only holds within simulation environment. In fact, despite of the wide acceptability of ns-2 in network community, it can't play the role of real world test bed completely.
- Constraints of actuators: As mentioned above, the codec used in this research is only a suggestion. The realization of such codec will accompany the practice of predictive control schemes in real world. Also, MPC's formulation does not count the codec's constraints into its objective function. Therefore, resultant controller is not really optimized for the corresponding codec.
- Complexity of Model Predictive Control: even without constraints, the MPC strategy is still too complicated that may prevent its implementation in real time control.

D. Future Work

Solving the research's limitations is a suggestion for future development. Specifically, future work can: (1) aim to find a real world test bed for practicing the MPC controllers, (2) solving the quadratic programming with the addition of constraints into MPC formulation and (3) seeking for fast methods of implementing the MPC strategies.

REFERENCES

[1] K. Shah, S. Bohacek and E. Jonckheere, "On predictability of data network traffic," *Proceedings of IEEE American Controls Conference*, Denver, Colorado, June 2003, pp. 1619-1624.

[2] H. Ohsaki, M. Murata and H. Miyahara, "Modeling end-to-end packet delay dynamics of the internet using system identification," *Proceedings of Seventeenth International Teletraffic Congress*, Salvador da Bahia, Brazil, December 2001, pp. 1027-1038.

[3] H. Ohsaki, M. Morita and M. Murata, "On modeling round-trip time dynamics of the internet using system identification," *Proceedings of the 2002 International Conference on Information Networking*, vol. 2343, January 2002, pp. 359-371.

[4] A. G. Parlos, "Identification of the internet end-to-end delay dynamics using multi-step neuro-predictors," *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 3, no. 3, May 2002, pp. 2460-2465.

[5] Q. P. Wang, D. L. Tan, N. Xi and Y. C. Wang, "The Control Oriented QoS: analysis and prediction," *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, Seoul, Korea, May 2001, pp. 1897-1920.

[6] Q. S. Jiang, R. Srinivasan and D. Slonowsky, "Measurement based traffic prediction using fuzzy logic," *IEEE CCECE Canadian Conference on Electrical and Computer Engineering*, vol. 2, August 2002, pp. 834-840.

[7] S. Doddi, “Empirical modeling of end-to-end delay dynamics in best-effort networks,” M.S. Thesis, Texas A&M University, College Station, TX, December 2003.

[8] E. S. Yu and C. Y. Roger Chen, “Traffic prediction using neural networks,” *Global Telecommunications Conference*, 1993, vol.2, pp.991-995, May 1993.

[9] A. Bhattacharya, “Flow control of real-time unicast multimedia applications in best-effort networks,” Ph.D. Dissertation, Texas A&M University, College Station, TX, May 2007.

[10] A. Bhattacharya, A. G. Parlos, “Flow Control of Real-time Multimedia Applications in Best-effort Networks”, *Recent Advances in Intelligent Control*, 2009, pp.311-356, February 2009.

[11] Y. Xia, D. Harrison, S. Kalyanaraman, K. Ramachandran and A. Venkatesan, “An accumulation-based congestion control model,” *IEEE International Conference on Communications*, Anchorage, Alaska, May 2003, pp. 657-663.

[12] Network simulator ns-2, University of California at Berkeley, CA. [Online]. Available: http://www.isi.edu/nsnam/ns/

[13] ITU-T Rec. P.800, “Methods for subjective determination of transmission quality,” August 1996.

[14] M. Kommaraju, “Predictor development for controlling real-time applications over the internet,” M.S. Thesis, Texas A&M University, College Station, TX, December 2005.

VITA

Name:	Thien Chi Duong
Address:	122-124 Bau Cat 2, Ward 12, Tan Binh District, Ho Chi Minh City, Vietnam
Email Address:	chithien1884@yahoo.com
Education:	B.S., Mechatronics, University of Technology at Ho Chi Minh City, 2007